

THE UNIVERSITY OF MICHIGAN  
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS  
Logic of Computers Group

Technical Report

LECTURES ON SWITCHING AND AUTOMATA THEORY

Calvin C. Elgot  
Research Associate, Willow Run Laboratories

UMRI Project 2755

under contract with:

ORDNANCE CORPS, DETROIT ORDNANCE DISTRICT  
CONTRACT NO. DA-20-018-ORD-16971, PROJECT NO. ORD R AND D (TB 2-001)  
DETROIT, MICHIGAN

administered by:

THE UNIVERSITY OF MICHIGAN RESEARCH INSTITUTE ANN ARBOR

January 1959

Emm

1m2

1243

## FOREWORD

At the time these lectures were being prepared and presented, the author was engaged in research supported by the Office of Naval Research Contract No. Nonr 1224(21) , and by the Office of Ordnance Research (Contract No. DA-20-018-ORD-16971). The lectures employ many concepts and theorems developed in the course of this research.

## TABLE OF CONTENTS

	Page
1. Introduction	1
2. Operations on Sets	2
3. Operations and Associated Relations	3
4. Lattices	6
5. Propositional Calculus and an Application	14
6. Multi-Terminal Networks	28
7. Automata and Sequential Circuits	35
8. Realization of Events	61
9. Bibliography	73

## 1. INTRODUCTION

These lectures include a study of two-terminal series-parallel relay contact networks, multi-terminal relay contact networks, and sequential networks. General, systematic techniques for the study of series-parallel networks have employed Boolean algebra<sup>58</sup> or propositional calculus.<sup>53</sup> A basic paper on multi-terminal relay contact nets makes use of matrices whose entries are elements of a Boolean algebra.<sup>39</sup> More recent work<sup>49,30,34,17</sup> makes use of the algebraic concepts: lattice, semi-group, group. The concept of ring when suitably specialized is intimately connected with Boolean algebras.<sup>6</sup> This indicates the extent to which algebra is invading the mathematical theory of switching. Much recent work<sup>18,26,31,32</sup> has employed more advanced aspects of logic than the propositional calculus. Indeed, the use of logic and its techniques has already produced significant results and promises still more fruitful ones.

We begin with some preliminary, simple, algebraic and logical concepts which will lead to a discussion of finite Boolean algebras, propositional calculus, and their application to switching theory.

The work of Lunts<sup>39</sup> (which lists results only) is discussed in Section 6. The arguments make use of an important correlation between matrices of zeros and ones and finite binary relations. This correlation is useful in other work on switching.<sup>19,23,35</sup>

Automata and sequential circuits are introduced in Section 7 and the connection with Burks-Wright logical nets<sup>25</sup> is pointed out. Moore's theory and related work of Mealy and Ginsburg<sup>33,34,45</sup> is expounded. The equivalence, in a certain reasonable sense, of finite automaton as defined here and as defined in Moore is indicated. Finite semi-groups are associated with automata in section 7.9. In the case of a permutation automaton<sup>33</sup> (backwards deterministic),<sup>23</sup> the associated semi-group is a group. An application is made of this association. Kleene's theory<sup>37</sup> of regularity as modified in item 28 of the bibliography is explained. An alternative notion of regularity is defined and its relation to the primary concept is established. The discussion terminates with an example illustrating several of the main results of the sections on automata theory.

These notes were the basis for the bulk of a course given by the author at The University of Michigan (1957, 1958) in the Department of Electrical Engineering.

## 2. OPERATIONS ON SETS

Let  $A$  be a set (class, collection, aggregate) of objects called elements (or points) of the set. "x is an element in A" is written " $x \in A$ ." For example, if  $\mathcal{J}$  is the set of positive integers,  $1 \in \mathcal{J}$ ,  $2 \in \mathcal{J}$ , etc. If  $A$  and  $B$  are sets, then  $A = B$  (i.e., "A" and "B" denote the same set) if and only if ( $x \in A \iff x \in B$ ). If  $A$  and  $B$  are sets, the union of  $A$  and  $B$ ,  $A \cup B$ , is defined as the set of all elements  $x$  such that  $x \in A$  or  $x \in B$  (i.e., in one or both); the intersection of  $A$  and  $B$ ,  $A \cap B$ , is the set of all  $x$  such that  $x \in A$  and  $x \in B$ .  $A \subseteq B$  means "if  $x \in A$ , then  $x \in B$ "; so does  $B \supseteq A$ .  $A \subset B$  ( $B \supset A$ ) means  $A \subseteq B$  but  $A \neq B$ .

Exercises. Prove that for all sets  $A, B, C$

- 2.1.  $A \cup A = A$
- 2.2.  $(A \cup B) \cup C = A \cup (B \cup C)$ ,
- 2.3.  $A \cap A = A$ ,
- 2.4.  $(A \cap B) \cap C = A \cap (B \cap C)$ ,
- 2.5.  $A \cap (A \cup B) = A$  [Proof.  $x \in (A \cap (A \cup B))$  if and only if  $x \in A$  and if  $x \in A \cup B$  if and only if  $x \in A$  and ( $x \in A$  or  $x \in B$ ) if and only if  $x \in A$  or ( $x \in A$  and  $x \in B$ ) if and only if  $x \in A$ .]
- 2.6.  $A \cup (A \cap B) = A$ ,
- 2.7. if  $A \subseteq B$ , then  $A \cup B = B$ ,
- 2.8. if  $A \cup B = B$ , then  $A \subseteq B$ ,
- 2.9. if  $A \subseteq B$ , then  $A \cap B = A$ ,
- 2.10. if  $A \cap B = A$ , then  $A \subseteq B$ ,
- 2.11.  $A \cup B = B \cup A$ ,
- 2.12.  $A \cap B = B \cap A$ ,
- 2.13.  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ ,
- 2.14.  $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ .
- 2.15. Example. If  $A = \{1, 2\}$ , i.e.,  $A$  consists of two (distinct) elements 1, 2, and  $B = \{2, 3\}$ , then  $A \cup B = \{1, 2, 3\}$  and  $A \cap B = \{2\}$ .
- 2.16. If  $A$  and  $B$  are sets with no elements in common, then  $A \cap B$  is undefined. In order to avoid this situation, it is customary to introduce the empty set, denoted by " $\emptyset$ ," which is defined to be a set with no elements, i.e.,  $x \notin \emptyset$  for all  $x$ . It follows  $A \cap B = \emptyset$  if  $A$  and  $B$  have no elements in common.  $\emptyset$  is the only set with no elements. Objects which are not sets also have no elements.
- 2.17. For all sets  $A$ , (1)  $A \cap \emptyset = \emptyset$ , (2)  $A \cup \emptyset = A$ .

### 3. OPERATIONS AND ASSOCIATED RELATIONS

Let  $S$  be an arbitrary set and  $*$  a binary operation defined over the set.

3.1. Let  $R_*$  be a binary relation on  $S$  defined as follows:  $xR_*y$  if and only if  $x = x * y$ . [" $xR_*y$ " is read "x is in the relation  $R_*$  to y."] Similarly define  $xR^*y$  if and only if  $y = x * y$ .

3.2. A relation  $R$  is reflexive over  $S$  if and only if  $xRx$  for every  $x \in S$ ; anti-symmetric if and only if  $xRy$  and  $yRx$  implies  $x = y$ ; and transitive if and only if  $xRy$  and  $yRz$  implies  $xRz$ .

3.3. Theorem. Let  $*$  be a binary operation on  $S$ . Then

- (a) if  $x * x = x$  for all  $x \in S$  i.e.,  $*$  is idempotent, then  $R_*$  is reflexive over  $S$ ;
- (b) if  $x * y = y * x$  for all  $x, y \in S$ , i.e.,  $*$  is commutative, then  $R_*$  is anti-symmetric;
- (c) if  $(x * y) * z = x * (y * z)$  for all  $x, y, z \in S$ , i.e.,  $*$  is associative, then  $R_*$  is transitive.

Proof.

- (a)  $xR_*x$  if and only if  $x = x * x$ .
- (b) If  $xR_*y$ , then  $x = x * y$ ; if  $yR_*x$ , then  $y = y * x$ . Since  $x * y = y * x$ , it follows  $xR_*y$  and  $yR_*x$  implies  $x = y$ .
- (c) Suppose  $xR_*y$  and  $yR_*z$ . To prove:  $xR_*z$ . Then (1)  $x = (x*y)$ , (2)  $y = (y * z)$ . From (1) it follows (3)  $x * z = (x * y) * z$  while from (2) follows (4)  $x * y = x * (y * z)$ . Since the operation is associative, (3) and (4) yield  $x * z = x * y$ , but  $x * y = x$  from (1) so that  $x * z = x$ , i.e.,  $xR_*z$ .

3.4. Statement 3.3 with  $R^*$  in place of  $R_*$  also holds.

3.5. Example.

- (1) Let  $\min$  be a binary operation on numbers (reals or integers or positive integers) defined as follows:  $\min(a,b) = a$  if and only if  $a \leq b$ ;  $\min(a,b) = b$  if and only if  $b \leq a$ . Then  $\min$  is idempotent, anti-symmetric, and transitive,  $aR_{\min}b$  if and only if  $a = \min(a,b)$  if and only if  $a \leq b$ ;  $aR^{\min}b$  if and only if  $b = \min(a,b)$  if and only if  $b \leq a$ . Thus  $R_{\min}$  is the relation  $\leq$  while  $R^{\min}$  is the relation  $\geq$ .
- (2) Let  $S$  be the set of rational numbers (fractions; let  $*$  be ordinary multiplication. Then:  $xR_*y$  if and only if  $x = x * y$  if and only if

$x = 0$  or  $y = 1$ . Therefore,  $0R_*x$  and  $xR_*1$  for all rational numbers  $x$ . Notice that multiplication of rational numbers is commutative and associative but not idempotent. Theorem 3.3 tells us that in this case  $R_*$  is anti-symmetric and transitive. Note, too,  $xR_*^*0$  and  $1R_*^*x$  for all rational numbers  $x$ .

(3) Let  $S$  be the set of non-zero rational numbers and let  $x * y$ . Then  $xR_*y$  if and only if  $x = x + y$  if and only if  $xy = x$  if and only if  $y = 1$ , while  $xR_*^*y$  if and only if  $y = x + y$  if and only if  $y^2 = x$  so that  $9R_*^*3$ ,  $16R_*^*4$ , etc.

### 3.6. Exercises.

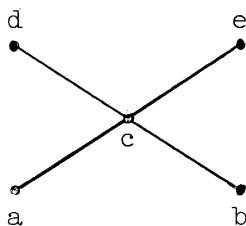
- (a) What is  $R_U$ ,  $R^U$ ,  $R_\cap$ ,  $R^\cap$  in terms of previously defined relations? [Note 2.1, 2.2, 2.11; 2.3, 2.4, 2.12.]
- (b) Show the greatest common divisor (gcd) and the least common multiple (lcm) of positive integers are idempotent, commutative, associative operations. What is  $R_*$  if  $x * y$  means gcd of  $x, y$ ; if  $x * y$  means lcm of  $x, y$ ?
- (c) If  $*$  is commutative, then  $xR_*y$  if and only  $yR_*^*x$ , and conversely.
- (d) Show subtraction of integers is not an associative nor a commutative operation.
- (e) If  $+$  is an associative binary operation on a set  $S$ , then: if  $xR^+m$  and  $yR^+m$ , then  $(x+y)R^+m$  (cf. 3.1). If, furthermore,  $+$  is commutative and idempotent, then  $xR^+x+y$  and  $yR^+x+y$ . Under these conditions  $(S, +)$  is a semi-lattice and  $R^+$  a semi-lattice ordering relation (cf. 3.3).

3.7. A partial ordering relation  $R$  over a set  $S$  is a reflexive, anti-symmetric and transitive relation over  $S$ .

#### Examples.

- (a)  $\subseteq, \supseteq$ .
- (b) Define  $R$  over ordered pairs [i.e.,  $(x, y)$  is to be distinguished from  $(y, x)$ ] of numbers (real, integers, etc.)  $(a, b)$  as follows:  $(a, b) R (c, d)$  if and only if  $a \leq c$  and  $b \leq d$ . Thus  $(2, 3) R (2, 4)$  but not  $(2, 3) R (3, 2)$  nor  $(3, 2) R (2, 3)$ .

(c)





$xRy$  if and only if (1)  $y$  is above  $x$  or (2)  $y = x$ .

(d)  $R_*$ ,  $R^*$  where  $*$  is idempotent, commutative, and associative.

3.8. Exercise.

(a) If  $R$  is a partial ordering relation over  $S$  and  $\overset{\cup}{R}$  is defined as follows:  
 $\overset{\cup}{R}xRy$  if and only if  $yRx$ , then  $\overset{\cup}{R}$  is a partial ordering relation.

(b)  $\overset{\cup}{R}xRy$  if and only if  $xRy$ .

## 4. LATTICES <sup>9,10</sup>

4.1. A lattice  $L = (A, \vee, \wedge)$  consists of a set  $A$  and two binary operations  $\vee, \wedge$  such that

- (a) each operation is idempotent, commutative, and associative and
- (b)  $a \wedge (a \vee b) = a$ ,  $a \vee (a \wedge b) = a$ , for all  $a, b$  in  $A$ . The first of the two operations is the join operation, while the second is the meet operation.

4.2. Exercise. If  $(A, \vee, \wedge)$  is a lattice, then so is  $(A, \wedge, \vee)$ .

4.3. Examples of lattices.

- (a) If  $A$  is the collection of all subsets of a set  $S$  ( $T$  is a subset of  $S$  iff  $T \subseteq S$ ), then  $(A, \cup, \cap)$  is a lattice. More generally, if  $A$  is a collection of sets such that  $A$  is "closed" under  $\cup$  (i.e.,  $S_1 \in A$  and  $S_2 \in A$  imply  $S_1 \cup S_2 \in A$ ) and  $\cap$ , then  $(A, \cup, \cap)$  is a lattice.
- (b) Let  $A$  be the set of ordered pairs of integers. Define:  $(a, b) \vee (c, d) = (\max(a, c), \max(b, d))$ ,  $(a, b) \wedge (c, d) = (\min(a, c), \min(b, d))$ . Then  $(A, \vee, \wedge)$  is a lattice.

4.4. Exercises.

- (1) Give 4.3(b) a geometric interpretation by introducing rectangular coordinates in a plane and correlating with an ordered pair of integers, the point which has this ordered pair as its coordinates.
- (2) If  $A$  is the set of positive integers, and if  $a \vee b$  means the least common multiple of  $a, b$ , and if  $a \wedge b$  means the greatest common divisor of  $a, b$ , then  $(A, \vee, \wedge)$  is a lattice.

4.5. A relation  $\leq$  is a lattice ordering relation over a set  $S$  iff  $\leq$  is a partial ordering relation and every pair of elements in  $S$  has a (unique) least upper bound and greatest lower bound, with respect to this ordering. ( $b$  is an upper bound of  $x, y$  means  $x \leq b$  and  $y \leq b$ .  $m$  is a least upper bound of  $x, y$  means that  $m$  is an upper bound and further if  $b$  is any upper bound of  $x, y$  then  $m \leq b$ .)

4.6. Theorem.

- (a) If  $(A, \vee, \wedge)$  is a lattice, then  $R^\vee = R^\wedge$  is a lattice ordering relation with  $\vee$  the least upper bound and  $\wedge$  the greatest lower bound with respect to this ordering.

---

\*Note: ff means "if and only if."

- (b) If  $\leq$  is any lattice ordering relation over a set  $A$ , let  $j(\leq)$  be the binary operation on  $A$  of least upper bound and let  $m(\leq)$  be the binary operation on  $A$  of greatest lower bound both with respect to  $\leq$ , then  $(A, j(\leq), m(\leq))$  is a lattice.

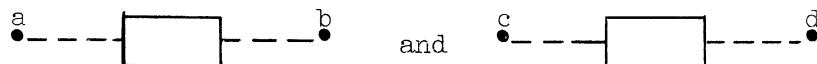
Proof.

- (a) Since  $\wedge$  is idempotent, commutative, and associative (4.1), it follows (3.3) that  $R_\wedge$  is a partial ordering relation. We note  $xR_\wedge y$  if and only if  $x = x \wedge y$ , while  $xR_\vee y$  if and only if  $y = x \vee y$  (3.1). If  $xR_\wedge y$ , then  $x \vee y = (x \wedge y) \vee y = y$  so that  $xR_\vee y$ . If  $xR_\vee y$ , then  $x \wedge y = x \wedge (x \vee y) = x$  so that  $xR_\wedge y$  (4.1(b)), which establishes  $R^\vee = R_\wedge$ . (To say  $R_1 = R_2$ , where  $R_1$  and  $R_2$  are binary relations, means that  $xR_1 y$  if and only if  $xR_2 y$ .) We now write " $\leq$ " instead of "R." Let  $x, y \in A$ . We note  $x \leq x \vee y$  if and only if  $x \vee y = x \vee (x \vee y)$ . But  $x \vee (x \vee y) = (x \vee (x \vee y))$ . But  $x \vee (x \vee y) = (x \vee x) \vee y = x \vee y$ . Similarly, since  $x \vee y = y \vee (x \vee y) = y \vee (y \vee x) = (y \vee x) = y \vee y \vee x = y \vee x$ , it follows  $y \leq x \vee y$ . Suppose now  $x, y \leq m$ . It is required to show that  $x \vee y \leq m$ , i.e.,  $m = (x \vee y) \vee m$ . Note that  $(x \vee y) \vee m = x \vee (y \vee m) = x \vee m$  (since  $y \leq m$ ) and  $x \vee m = m$  (since  $x \leq m$ ) so that  $x \vee y \leq m$ .

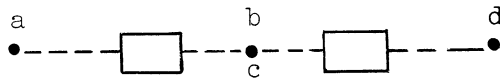
4.7. Exercise.

- (a) Prove 4.6(b).
- (b) Define  $aRb$ , where  $a, b$  are positive integers as follows:  $aRb$  if and only if there exists an (integer)  $x$  such that  $a \cdot x = b$ , where " $\cdot$ " indicates ordinary multiplication, i.e.,  $aRb$  if and only if  $a$  divides  $b$ . Verify that  $R$  is a lattice ordering relation and determine the arithmetic meaning of join and meet in the associated lattice. [Cf. 4.1, 4.5, 4.6.]
- (c) Every finite lattice has a (unique) maximum and a unique minimum element, with respect to the lattice ordering.
- (d) If  $\leq$  is a partial ordering relation over a set  $S$  such that for any  $a, b \in S$  either  $a \leq b$  or  $b \leq a$ , then  $\leq$  is a lattice ordering. (A relation satisfying these properties is called a complete, simple, or linear ordering relation.)

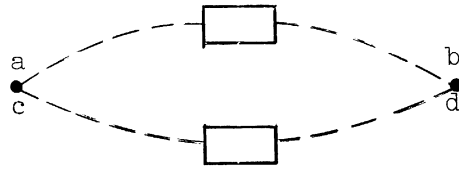
4.8. The meaning of series-parallel network is now indicated in a semi-formal way. If



are two-terminal networks, their connection in series is

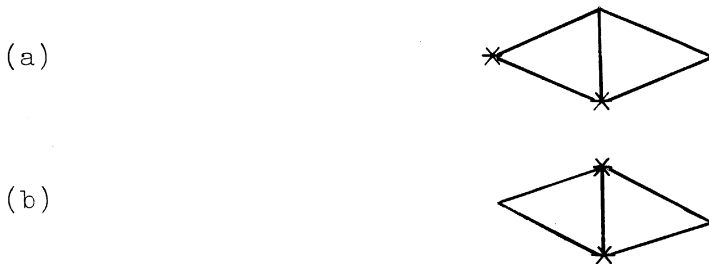


their connection in parallel is



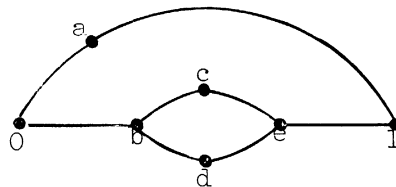
A network,  $a \xrightarrow{\quad} b$ , consisting of two nodes (vertices, points) and one branch (edge) is called a unit network. The two terminals of each of the networks above are the displayed ones, the first one being on the left. The class of series-parallel networks is the smallest class of two-terminal networks containing the unit networks and such that the series and parallel connections of any two networks in the class is again in the class. Equivalently, a two-terminal network is series-parallel if and only if it is a unit network or is obtainable from unit networks by a finite number of series and parallel connections.

4.9. Exercise. Verify, using the definition, that the following networks, where the two terminals are indicated by "\*", are series-parallel.



4.10. By a chain in a two-terminal network we shall mean a sequence of distinct nodes beginning with the first terminal and ending with the second having the further property that consecutive nodes have an edge in common. A relation  $<$  may now be defined on the nodes of the network as follows:  $x < y$  if and only if  $x$  is a node which precedes  $y$  in some chain containing both. It turns out that: if the network is series-parallel, then  $\leq$  (where  $<$  is the relation just defined) is a lattice ordering of the nodes of the network. In accordance with 4.6(b), a lattice may be associated with a series-parallel network.

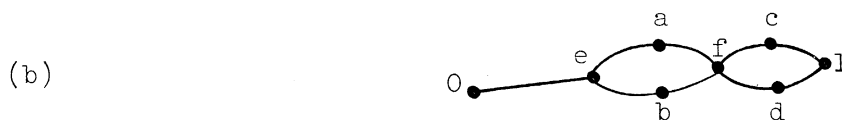
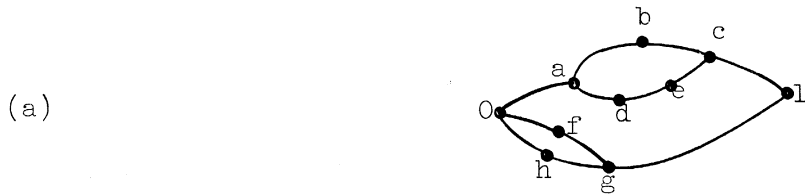
For example, if the network is



then  $b \leq c$ ,  $b \leq e$ ,  $b \not\leq a$ ,  $c \not\leq a$ .

4.11. Two elements  $x, y$  in a lattice with  $0, 1$  ( $0$  is the minimum and  $1$  is the maximum element with respect to  $\leq$ ) are called complementary if and only if  $x \vee y = 1$  and  $x \wedge y = 0$ . In the example above (4.10) the following pairs are complementary:  $\{a, c\}$ ,  $\{a, b\}$ ,  $\{a, d\}$ ,  $\{a, e\}$ ,  $\{0, 1\}$ . On the other hand, if the network above is series-connected with a unit network, then none of the elements in the associated lattice, except  $0, 1$  has complements.

4.12. Exercises. Construct a table for  $\vee$  and one for  $\wedge$  (analogous to addition and multiplication tables) for the lattice associated with the networks below. List the complementary pairs of elements.



4.13. A two-terminal network is admissible if and only if each node occurs in some chain (cf. 4.10). The following can be proved:<sup>30</sup> an admissible network is series-parallel if and only if  $<$  (the relation defined in 4.10) is asymmetric, that is, for no nodes  $x, y$  does  $x < y$  and  $y < x$  hold. It seems reasonable to call an admissible two-terminal network a bridge network if and only if  $<$  is not asymmetric. It follows from the theorem and definitions that every two-terminal network falls into one and only one of the following categories:

1. series-parallel,
2. bridge,
3. nonadmissible.

Exercise. Give two examples of each of these categories of networks.

4.14. A lattice  $(A, \wedge, \vee)$  is called distributive if and only if for all  $x, y, z \in A$  the following hold:  $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ , and  $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ .

4.15. In a distributive lattice every element has at most one (possibly no) complement. This assertion is an immediate consequence of the following theorem.

4.16. Theorem. In a distributive lattice, if  $a \vee x = a \vee y$ , and  $a \wedge x = a \wedge y$ , then  $x = y$ .

Proof. Note that:  $x = x \wedge (a \vee x) = x \wedge (a \vee y) = (x \wedge a) \vee (x \wedge y) = ((x \wedge a) \vee x) \wedge ((x \wedge a) \vee y) = x \wedge ((y \wedge a) \vee y) = x \wedge y$ , so that  $x = x \wedge y$ . If  $x$  and  $y$  are interchanged throughout the above argument, it remains valid so that  $y = y \wedge x$ . This together with  $x = x \wedge y$  yields  $x = y$ .

4.17. In the diagram of 4.10,  $b, c, d, e$  are each complements of  $a$ . It follows from 4.15 that the lattice associated with diagram 4.10 is not distributive.

4.18. A distributive lattice with 0,1 in which each element has a complement (by 4.15 unique) is a Boolean algebra.

Examples.

- (1) The set of all subsets of a set (finite or infinite) under union and intersection. Given the set  $\{a, b, c\}$  with three elements, the set of all subsets is  $\{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$ . Notice that the given set has three elements while the set of all subsets has eight elements. In general, if a set has  $n$  elements, the set of all subsets has  $2^n$  elements. This may be proved by putting subsets of an  $n$ -element set into one-to-one correspondence with all the sequences of zeros and ones of length  $n$ . In case  $n = 3$  and the given set is  $\{a, b, c\}$ , a correspondence is:

000	$\emptyset$
100	$\{a\}$
010	$\{b\}$
001	$\{c\}$
110	$\{a, b\}$
101	$\{a, c\}$
011	$\{b, c\}$
111	$\{a, b, c\}$ .

Another count of the number of subsets of a given set is obtainable as follows. According to the binomial theorem

$$(a + b)^n = \sum_{r=0}^n \binom{n}{r} a^{n-r} b^r ,$$

where  $\binom{n}{r}$  is the number of  $n$  things taken  $r$  at a time, i.e.,  $\binom{n}{r}$  is the number of  $r$ -element subsets of an  $n$ -element set. Substituting in the formula  $a = 1$ ,  $b = 1$  yields

$$2^n = \sum_{r=0}^n \binom{n}{r} .$$

(2) Let  $n$  be a positive integer which is not divisible by the square of any number except 1. Let  $A$  be the set of all positive integral divisors of  $n$ , and let  $x \vee y$  be the least common multiple of  $x, y$  and let  $x \wedge y$  be the greatest common divisor of  $x, y$ . Then  $(A, \vee, \wedge)$  is a Boolean algebra. We indicate the proof in the case that  $n = 30$ . In this case,  $A = \{1, 2, 3, 5, 6, 10, 15, 30\}$ . Let  $f$  be a function defined as follows:

$$\begin{array}{ll} f(\emptyset) = 1 & f(\{2, 3\}) = 6 \\ f(\{2\}) = 2 & f(\{2, 5\}) = 10 \\ f(\{3\}) = 3 & f(\{3, 5\}) = 15 \\ f(\{5\}) = 5 & f(\{2, 3, 5\}) = 30. \end{array}$$

Then

- (a)  $f(x \cup y) = f(x) \vee f(y)$ ,
- (b)  $f(x \cap y) = f(x) \wedge f(y)$ , and
- (c) for every  $v \in A$ , there is a subset  $u$  of  $\{x, y, z\}$  such that  $v = f(u)$ .

Because of (a), (b), (c) and the fact that all the subsets of a set under  $\cup, \cap$  form a Boolean algebra, it follows that  $(A, \vee, \wedge)$  is a Boolean algebra.

4.19. If  $f$  is a function defined on a set  $A$  with values in a set  $A'$  and if  $\mathcal{A} = (A, \vee, \wedge)$ ,  $\mathcal{A}' = (A', \vee', \wedge')$  where  $\vee, \wedge, \vee', \wedge'$  are binary operations such that

- (1)  $f(a \vee b) = f(a) \vee' f(b)$ ,
- (2)  $f(a \wedge b) = f(a) \wedge' f(b)$ ,

then  $f$  is a homomorphism of  $\mathcal{A}$  into  $\mathcal{A}'$ . If, furthermore,

- (3) for every  $x' \in A'$ , there is an  $x \in A$  such that  $f(x) = x'$ ,

then  $f$  is a homomorphism of  $\mathcal{A}$  onto  $\mathcal{A}'$ . If, in addition,

- (4)  $x \neq y$  implies  $f(x) \neq f(y)$ ,

then  $f$  is an isomorphism of  $\mathcal{A}$  onto  $\mathcal{A}'$ .

The function  $f$  of 4.18 (2) is an isomorphism.

#### 4.20. Exercises.

- (1) If  $f$  is a homomorphism from a lattice  $\mathcal{A} = (A, \vee, \wedge)$  onto  $\mathcal{A}' = (A', \vee', \wedge')$  (where  $\vee', \wedge'$  are binary operations), then  $(A', \vee', \wedge')$  is a lattice. If  $\mathcal{A}$  is a Boolean algebra, then  $\mathcal{A}'$  is a Boolean al-

gebra, and if  $a \in A$  and  $b$  is the (unique) complement of  $a$ , then  $f(b)$  is the complement of  $f(a)$ .

(2) Define operations  $\vee, \wedge$  on the ordered triples of 4.18 (1) in such a way that the correspondence indicated between these triples and the subsets of  $\{a,b,c\}$  is an isomorphism.

(3) Let  $S$  be any set (finite or infinite). Let  $A$  be the class of all subsets  $x$  such that either  $x$  is a finite subset of  $S$  or  $x = S - y$  and  $y$  is a finite subset of  $S$ ; here  $S - y$  is the set of all elements in  $S$  which are not in  $y$ . Show that  $(A, \cup, \cap)$  is a Boolean algebra.

4.21. We now establish some identities for Boolean algebras. The (unique) complement of an element  $x$  in a Boolean algebra will be denoted by " $\bar{x}$ ."

$$(1) \overline{x \vee y} = \bar{x} \wedge \bar{y}.$$

Proof: Using the left distributive law for  $\vee$  over  $\wedge$ :  $(x \vee y) \vee (\bar{x} \wedge \bar{y}) = [(x \vee y) \vee \bar{x}] \wedge [(x \vee y) \vee \bar{y}] = 1 \wedge 1 = 1$ . Similarly using the right distributive law for  $\wedge$  over  $\vee$ :  $(x \vee y) \vee (\bar{x} \wedge \bar{y}) = [x \wedge (\bar{x} \wedge \bar{y})] \vee [y \wedge (\bar{x} \wedge \bar{y})] = 0 \vee 0 = 0$ .

$$(2) \overline{\bar{x} \wedge \bar{y}} = x \vee y.$$

Proof: Interchange  $\vee, \wedge$  and  $0, 1$  in the argument for (1).

$$(3) \bar{\bar{x}} = x.$$

Proof:  $\bar{\bar{x}}$  is, by definition, the unique complement of  $\bar{x}$ . But  $x$  and  $\bar{x}$  are complementary, i.e.,  $x$  is the unique complement of  $\bar{x}$ . Therefore  $x = \bar{\bar{x}}$ .

$$(4) \text{ If } x \leq y \text{ then } \bar{y} \leq \bar{x}.$$

Proof: We note that  $x \leq y$  if and only if  $x \vee y = y$  if and only if  $x \wedge y = x$  and that  $\bar{y} \leq \bar{x}$  if and only if  $\bar{y} \vee \bar{x} = \bar{x}$  if and only if  $\bar{y} \wedge \bar{x} = \bar{y}$ . Since by hypothesis  $x \leq y$ , it follows that  $\bar{y} = \overline{x \vee y}$  and, by (1),  $\bar{y} = \bar{x} \wedge \bar{y}$ , so that  $\bar{y} \leq \bar{x}$ .

$$(5) \text{ If } \bar{y} \leq \bar{x} \text{ then } x \leq y.$$

Proof: By (4),  $\bar{\bar{x}} \leq \bar{\bar{y}}$  and, by (3),  $x \leq y$ .

#### 4.22. Exercises.

(1) Let  $A$  be the set of positive integers augmented by a new element  $\infty$  (infinity). If  $x, y \in A$ , define  $x \vee y$  and  $x \wedge y$  as in 4.4 (2) if  $x, y$  are both integers; define  $x \vee \infty = \infty$ ,  $x \wedge \infty = x$  for all  $x \in A$ .

(a) Determine whether  $(A, \vee, \wedge)$  is a lattice.

(b) Which elements, if any, have complements?

(c) Is  $(A, \vee, \wedge)$  a Boolean algebra?



- (2) Show, by mathematical induction, that  $\overline{x_1 \vee x_2 \vee \dots \vee x_n} = \bar{x}_1 \wedge \bar{x}_2 \wedge \dots \wedge \bar{x}_n$ , for any  $n \geq 1$ .
- (3) In a Boolean algebra  $(A, \vee, \wedge)$  define  $x + y = (x \wedge \bar{y}) \vee (\bar{x} \wedge y)$ . Show:
- (a)  $(x + y) + z = x + (y + z)$ ;
  - (b)  $x + x = 0$ ;
  - (c)  $x \wedge (y + z) = (x \wedge y) + (x \wedge z)$ ;
  - (d)  $1 + x = \bar{x}$ ;
  - (e)  $x \vee y = x + y + (x \wedge y)$ ;
  - (f) let  $A$  be a class of subsets of  $S$  closed under union and intersection, and interpret  $\vee, \wedge$  as union, intersection, respectively; interpret  $\bar{x}$  and  $x + y$  where  $x, y \in A$ .
- (4) Let  $A$  be the set  $\{0,1\}$ ;  $0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1$ ,  $0 \vee 0 = 0$ ;  $0 \wedge 1 = 1 \wedge 0 = 0 \wedge 0 = 0$ ,  $1 \wedge 1 = 1$ . Show  $(A, \vee, \wedge)$  is a Boolean algebra.
- (5) Let  $B$  be the set of all functions  $f$  defined on  $n$ -tuples of 0's and 1's with values which are 0 or 1. Define  $f \vee g = h$  if and only if  $f(x_1, x_2, \dots, x_n) \vee g(x_1, x_2, \dots, x_n) = h(x_1, x_2, \dots, x_n)$  where each  $x_i$  varies independently over 0,1 and the " $\vee$ " operation on 0,1 is defined as in (4). Define  $f \wedge g$  analogously. Show  $(B, \vee, \wedge)$  is a Boolean algebra. How many elements does  $B$  have?

5. PROPOSITIONAL CALCULUS AND AN APPLICATION<sup>10-15,58</sup>

5.1. A formula is a finite (nonzero) concatenation of symbols of the following four types

propositional variables:  $p_1, p_2, \dots, p_n$        $[n \geq 1]$   
 propositional constants:  $f, t$   
 propositional connectives:  $\vee, \wedge, \sim$   
 punctuation symbols:  $(, )$ .

For example,  $p_1 p_2 \vee (, \sim p_1 p_2$  are formulas. Such formulas will not be interesting to us nor will they be given interpretations. The formulas with which we shall be concerned and to which interpretations will be given are the well-formed formulas. We define inductively on the length (number of occurrences of symbols) of a formula, a well-formed formula. A formula  $F$  is a well-formed formula if and only if

- (a)  $F$  is a propositional variable or constant;
- (b)  $F$  is  $(F_1 \vee F_2)$  or  $(F_1 \wedge F_2)$  or  $(\sim F_1)$  where  $F_1$  and  $F_2$  are well-formed formulas.

The symbols  $\vee, \wedge, \sim, f, t$  will be read "or," "and," "not," "falsehood," "truth," respectively.

5.2. (a) We may allow an infinite number of propositional variables in 5.1 but for greater perspicuity in the applications to switching theory the finite cases will be used.

(b) The punctuation symbols of 5.1 are dispensable. We may, for example, write  $\vee p_1 p_2$  instead of  $(p_1 \vee p_2)$ ,  $\wedge p_1 p_2$  instead of  $(p_1 \wedge p_2)$ ,  $\sim p_1$  instead of  $(\sim p_1)$ , so that  $(p_1 \vee (\sim(p_1 \wedge p_2)))$  may be written  $\vee p_1 \sim \wedge p_1 p_2$ . We shall, however, adhere to the scheme of 5.1 with the modification, however, that parentheses will sometimes be dropped when it is believed that this will not cause confusion. The syntax as well as some applications of parenthesis-free notation are discussed in items 24 and 29 of the bibliography.

5.3. Define  $t \sqcup t = t \sqcup f = f \sqcup t = t, f \sqcup f = f$ ;  
 $t \sqcap t = t$  and  $t \sqcap f = f \sqcap t = f \sqcap f = f$  [cf. 4.22 (4)];  
 $\square t = f, \square f = t$ .

We will now associate with each well-formed formula  $F$  a function  $g_F^{(n)}$  from  $n$ -tuples of  $t$ 's and  $f$ 's and with values which are  $t$  or  $f$  as follows (the  $x_i$ 's below vary independently through  $\{t, f\}$ ):

- (1)  $g_F^{(n)}(x_1, x_2, \dots, x_n) = x_i$  if  $F$  is  $p_i$ ;  
 $g_F^{(n)}(x_1, x_2, \dots, x_n) = t$  if  $F$  is  $t$ ;  
 $g_F^{(n)}(x_1, x_2, \dots, x_n) = f$  if  $F$  is  $f$ ;
- (2)  $g_F^{(n)}(x_1, x_2, \dots, x_n) = g_{F_1}^{(n)}(x_1, x_2, \dots, x_n) \cup g_{F_2}^{(n)}(x_1, x_2, \dots, x_n)$  if  $F$  is  $(F_1 \vee F_2)$ ;
- (3)  $g_F^{(n)}(x_1, x_2, \dots, x_n) = g_{F_1}^{(n)}(x_1, x_2, \dots, x_n) \cap g_{F_2}^{(n)}(x_1, x_2, \dots, x_n)$  if  $F$  is  $(F_1 \wedge F_2)$ ;
- (4)  $g_F^{(n)}(x_1, x_2, \dots, x_n) = \neg g_{F_1}^{(n)}(x_1, x_2, \dots, x_n)$  if  $F$  is  $(\sim F_1)$ .

Now define two formulas  $F, G$  to be equivalent,  $F \stackrel{n}{\text{eq}} G$ , if  $g_F^{(n)} = g_G^{(n)}$ .

5.4. Theorem.  $F \stackrel{n}{\text{eq}} G$ , where at most the variables  $p_1, \dots, p_n$  occur in  $F, G$ , if and only if  $F \stackrel{m}{\text{eq}} G$  where  $m > n$ .

Proof. It is obvious by induction on the number of occurrences of operation symbols ( $\vee, \wedge, \sim$ ) in  $F$  that  $g_F^{(m)}(x_1, x_2, \dots, x_n, \dots, x_m) = g_F^{(n)}(x_1, x_2, \dots, x_n)$  from which the result follows.

Because of this theorem we write " $F \text{ eq } G$ " if  $F \stackrel{n}{\text{eq}} G$  for some  $n$ .

5.5(a). A binary relation  $R$  which over a set  $S$  satisfies:

- (1)  $aRa$  for every  $a \in S$  (reflexivity)
- (2)  $aRb$  implies  $bRa$  (symmetry)
- (3)  $aRb$  and  $bRc$  imply  $aRc$  (transitivity)

is called an equivalence relation. Show that if  $R^*(a)$  is the set of all  $x$  such that  $xRa$ , then

- (4)  $a \in S, b \in S$ , and  $aRb$  imply  $R^*(a) = R^*(b)$
- (5)  $a \in S, b \in S$ , and  $\neg aRb$  imply  $R^*(a) \cap R^*(b) = \emptyset$
- (6)  $S$  is the union of all classes  $R^*(a)$  where  $a$  runs through  $S$ .

5.5(b) Show  $\stackrel{n}{\equiv}$  is an equivalence relation over the integers for each positive integer  $n$ , where  $a \stackrel{n}{\equiv} b$  means  $a-b$  is divisible by  $n$

Conversely, if  $S$  is the union of disjoint subclasses (i.e., if  $U, V$  are subclasses either  $U = V$  or  $U \cap V = \emptyset$ ), and if  $aRb$  is interpreted as meaning that  $a, b$  are both in the same subclass, then  $R$  satisfies (1), (2), (3).

5.6. The class of all well-formed formulas  $W$  may be partitioned into subclasses  $W_1, \dots, W_r$  (cf. 5.5) such that any two formulas in the same subclass are eq-related,  $W_i \cap W_j = \emptyset$  for  $i \neq j$  and  $W = W_1 \cup \dots \cup W_r$ . Note that if  $F \text{ eq } F'$

and  $G \text{ eq } G'$ , then  $(F \vee G) \text{ eq } (F' \vee G')$  and  $(F \wedge G) \text{ eq } (F' \wedge G')$ . Hence we may define  $W_k = W_i \vee W_j$  if  $H$  is  $F \vee G$  and  $H \in W_k, F \in W_i, G \in W_j$ .  $W_i \wedge W_j$  is defined analogously.

Theorem. If  $C = \{W_1, \dots, W_r\}$ , then  $(C, \vee, \wedge)$  is a Boolean algebra where  $\vee, \wedge$  are the operations defined immediately above. Indeed  $(C, \vee, \wedge)$  is isomorphic to  $(B, \vee, \wedge)$  of 4.22 (5).

Proof. By virtue of 5.3 and the definition of eq, with each  $W_i$  is associated a unique function from  $n$ -tuples of  $t$ 's and  $f$ 's with values which are  $t$  or  $f$ . Moreover, with each function  $g$  of this kind there is a formula  $F$  such that  $g_F^{(n)} = g$ . The formula  $F$  may be chosen as a disjunction of conjunctions of the form  $q_1 \wedge q_2 \wedge \dots \wedge q_n$  where each  $q_i$  is  $p_i$  or  $(\sim p_i)$ ; for each  $n$ -tuple  $x_1, x_2, \dots, x_n$  such that  $f(x_1, x_2, \dots, x_n) = t$  construct the "corresponding" conjunction of  $q_i$ 's where  $q_i = p_i$  if and only if  $x_i = t$ ; the disjunction of all conjunctions of this kind "corresponding" to  $n$ -tuples  $x_1, x_2, \dots, x_n$  such that  $f(x_1, x_2, \dots, x_n) = t$  is chosen as  $F$ . For this  $F$ ,  $g_F^{(n)} = g$ . Hence, for each function  $g$  from  $n$ -tuples of  $t$ 's and  $f$ 's with values which are  $t$  or  $f$ , there is a  $W_i$  such that for each  $F \in W_i$ ,  $g_F^{(n)} = g$ . The correspondence indicated between  $C$  and these functions  $g$  yields a correspondence between  $C$  and  $B$  of 4.22 (5) if  $t$  is correlated with 1 and  $f$  with 0. The correspondence between  $C$  and  $B$  is an isomorphism between  $(C, \vee, \wedge)$  and  $(B, \vee, \wedge)$  of 4.22 (5). Since the latter system is a Boolean algebra, the former is. Note, too, this implies  $r = 2^{2^n}$ .

5.7. By virtue of the theorem of 5.6, the identities which hold for Boolean algebras carry over to formulas of any propositional calculus with "=" replaced by "eq." For example,  $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$  implies  $F \wedge (G \vee H) \text{ eq } (F \wedge G) \vee (F \wedge H)$ , and  $\overline{x \vee y} = \overline{x} \wedge \overline{y}$  implies  $\sim(F \vee G) \text{ eq } (\sim F) \wedge (\sim G)$  where  $F, G, H$  are arbitrary well-formed formulas of a propositional calculus.

5.8. If  $G$  is a well-formed part of a well-formed formula  $F$  and  $G \text{ eq } G'$ , then  $F \text{ eq } F'$  where  $F'$  is obtained from  $F$  by replacing an occurrence of  $G$  by  $G'$ . This may be proved by induction on the number of operation symbols which occur in  $F$ . For example, if  $F$  is

$$(p_1 \vee (p_2 \wedge (\sim p_2))) \wedge p_3 \quad ,$$

then  $G$  may be taken as  $p_2 \wedge (\sim p_2)$ ,  $G'$  as 0, so that  $F'$  is  $(p_1 \vee 0) \wedge p_3$ . Repeating the process with " $G$ " the formula  $(p_1 \vee 0)$  and " $G'$ " the formula  $p_1$ , we obtain  $p_1 \wedge p_3 \text{ eq } (p_1 \vee 0) \wedge p_3$ . Thus

$$(p_1 \vee (p_2 \wedge (\sim p_2))) \wedge p_3 \text{ eq } p_1 \wedge p_3 \quad .$$

5.9. Exercises. Let  $A, B, C$  be well-formed formulas of a propositional calculus. Define:

$$\begin{aligned} (A \equiv B) & \text{ to be } (A \wedge B) \vee ((\sim A) \wedge (\sim B)) \quad ; \\ (A + B) & \text{ to be } (A \wedge (\sim B)) \vee ((\sim A) \wedge B) \quad ; \end{aligned}$$

- $(A \rightarrow B)$  to be  $(\sim A \vee B)$  ;  
 $(A \uparrow B)$  to be  $(\sim(A \wedge B))$  ;  
 $(A \downarrow B)$  to be  $(\sim(A \vee B))$  .

Show that

- (1)  $(t + A) \text{ eq } (\sim A)$ ;
- (2)  $((A + B) + C) \text{ eq } (A + (B + C))$ ;
- (3)  $((A \equiv B) \equiv C) \text{ eq } (A \equiv (B \equiv C))$ ;
- (4)  $(A + B) \text{ eq } (B + A)$  and  $(A \equiv B) \text{ eq } (B \equiv A)$
- (5)  $(t + A + B) \text{ eq } (A \equiv B)$ ;
- (6)  $((A \rightarrow B) \wedge (B \rightarrow A)) \text{ eq } (A \equiv B)$ ;
- (7)  $(A \rightarrow (B \rightarrow A)) \text{ eq } t$ ;
- (8)  $A \text{ eq } B$  if and only if  $(A \equiv B) \text{ eq } t$ ;
- (9)  $\text{Eq}(A) \leq \text{Eq}(B)$  if and only if  $(A \rightarrow B) \text{ eq } t$ , where  $\text{Eq}(A)$  is the class of formulas  $X$  such that  $X \text{ eq } A$  and  $\leq$  is the lattice ordering of the Boolean algebra (cf. Theorem of 5.6);
- (10) there is a formula in  $A, B, +, \wedge, t$  which is eq-related to
  - (a)  $A \vee B$
  - (b)  $A \equiv B$
  - (c)  $\sim A$ ;
- (11) same as (10) but replace "+,  $\wedge, t$ " by " $\rightarrow, f$ ";
- (12) same as (10) but replace "+,  $\wedge, t$ " by " $\uparrow$ ";
- (13) same as (10) but replace "+,  $\wedge, t$ " by " $\downarrow$ ";
- (14)  $A \vee (A \wedge B) \text{ eq } A$ ;
- (15)  $A \vee (\bar{A} \wedge B) \text{ eq } A \vee B$  [cf. 5.10];
- (16)  $(A \wedge B) \vee (\bar{A} \wedge C) \text{ eq } (A \wedge B) \vee (\bar{A} \wedge C) \vee (B \wedge C)$ .

5.10. Notice that  $A \equiv B \equiv C$  (i.e.,  $((A \equiv B) \equiv C)$ ) is not eq-related to  $((A \equiv B) \wedge (B \equiv C))$ . Note, too, that  $\equiv, \rightarrow$  are operations while eq,  $\leq$  are relations (cf. (8), (9) of 5.9). Furthermore  $(A \equiv B)$  is (an abbreviation for) a well-formed formula of the propositional calculus but  $(A \text{ eq } B)$  is not.

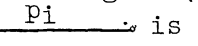
In what follows we will often regard a formula as denoting its equivalence class, so that the formula will denote an element of the Boolean algebra of 5.6. Under these circumstances we may write " $A = B$ " instead of " $A \text{ eq } B$ " and " $\bar{A}$ " instead of " $(\sim A)$ ."

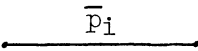
5.11. Application to series-parallel relay contact circuits. We shall restrict the meaning of relay contact to a device with two terminal posts which is capable of two states, in one of which there is a closed path between the two posts and in the other of which there is no closed path between the two posts. A contact is normally open if and only if there is no closed path between its posts when its controlling coil is not energized and there is a closed path between its posts in the contrary case. A contact is normally closed if and only if there is a closed path between its posts when its controlling coil is not energized and there is no closed path between its posts in the contrary case.

With a two-terminal series-parallel circuit constructed of relay contacts, but not their coils, we associate a diagram of the type indicated in 4.8 which indicates the structure of the circuit with the modification that each branch will be labeled by exactly one of the following:  $p_1, \bar{p}_1, p_2, \bar{p}_2, \dots, p_n, \bar{p}_n$  and in such a way that

(a) a letter without a bar is correlated with a normally open contact; a letter with a bar is correlated with a normally closed contact;

(b) if one branch is labeled  $q_i$  ( $q_i$  is  $p_i$  or  $\bar{p}_i$ ) and another  $q_j$ , then  $i = j$  if and only if the coils controlling the contacts associated with  $q_i, q_j$  are the same.

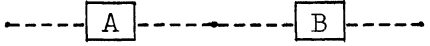
We now associate with each 4.8 diagram (but with branches labeled) a formula as follows. With the diagram  is associated the formula  $p_i$  which will be interpreted as "coil  $p_i$  is energized." (By coil  $p_i$  we mean the coil which controls the contact associated with the branch labeled " $p_i$ ." ) With the diagram



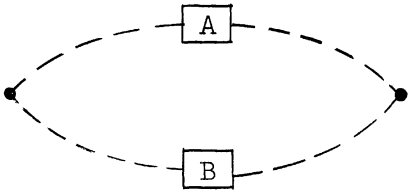
is associated the formula  $\bar{p}_i$  which will be interpreted as "coil  $p_i$  is not energized." Labels "t", ("f") on such a diagram are interpreted as "there is a closed (open) path between the terminals." If with the diagrams



the formulas A, B are respectively associated, then with their series connection

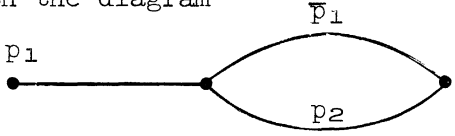


the formula  $(A \wedge B)$  is associated, and with their parallel connection



the formula  $(A \vee B)$  is associated. The interpretation of  $(A \wedge B)$  is obtained by following the interpretation of A by "and" and then the interpretation of B. The interpretation of  $(A \vee B)$  is obtained similarly but writing "or" instead of "and." Here "or" is understood in the inclusive sense, i.e.,  $(A \vee B)$  is false if and only if both A and B are false.

For example, with the diagram



is associated the formula  $p_1 \wedge (\bar{p}_1 \vee p_2)$  which is interpreted as: coil  $p_1$  is energized and (either) coil  $p_1$  is not energized or coil  $p_2$  is energized.

Conversely, given any well-formed formula constructed out of  $p_1, \bar{p}_1, p_2, \bar{p}_2, \dots$  by means of  $\vee, \wedge$ , there is a two terminal series-parallel network (essentially unique) which has the given formula as its associated formula. Notice that there is a closed path between the terminals of a two-terminal network when and only when the associated formula (under the given interpretation) is true.

Two networks are equivalent if and only if there is a closed path between the terminals of one when and only when there is a closed path between the terminals of the other. If this statement is properly understood, it follows that two networks  $C_1, C_2$  are equivalent if and only if their associated formulas  $F_1, F_2$  are eq-related, i.e.,  $F_1 \text{ eq } F_2$ . Thus the network indicated above is equivalent to the network.



since  $(p_1 \wedge (\bar{p}_1 \vee p_2)) \text{ eq } (p_1 \wedge p_2)$ . The former network contains three relay contacts while the latter network contains only two.

5.12. Exercises. (1) Let  $P = a \vee (\bar{a} \wedge b \wedge \bar{c}) \vee (\bar{a} \wedge \bar{b} \wedge c) \vee (\bar{a} \wedge \bar{b} \wedge \bar{c})$ .

(a) Write  $P$  in expanded disjunctive normal form (i.e., as a disjunction of conjunctions, each conjunction of which involves all three letters).

(b) Put the negation (complement) of  $P$  in the same form. [This may be done by inspection from (a).]

(c) Simplify  $P$  and indicate with a diagram the two-terminal series-parallel network associated with the simplified formula.

(2) The formula  $(a \wedge b) \vee (a \wedge c) \vee (a \wedge b) \vee (b \wedge c)$  is a disjunction of four conjunctions. Which, if any, of the conjunctions, when deleted, yields a formula equivalent to the given one?

(3) Find a relay contact network equivalent to the one given below but using at most five relay contacts. Hint: The result of (2) may be useful.

(4) Construct a relay contact network (if one exists) controlled by three coils and such that whenever the state of exactly one of the coils changes, the state of the network changes.

(5) Same as (4) but change "exactly one" to "exactly two."





- Rules:
- (1) If  $A, B$  are clauses of a formula and  $A$  subsumes  $B$ , then delete  $A$  from the formula.
  - (2) If  $p_i$  and  $\bar{p}_i A$  (for some  $i$ ) are clauses of a formula, replace  $\bar{p}_i A$  by  $A$ ; similarly if  $p_i$  and  $\bar{p}_i$  are interchanged.
  - (3) If  $p_i A$  and  $\bar{p}_i B$  are clauses of a formula  $G$  and  $A$  and  $B$  are both nonempty, then replace  $G$  by  $G \vee C$ , where  $C$  is obtainable from  $AB$  by deleting duplications of literals; this rule is applicable provided that (a) no letter both with and without a bar occurs in  $AB$  and (b)  $C$  subsumes no conjunction which is part of  $G$ .

Proof. Note that a clause of a formula is an implicant of that formula. To show the rules can be applied only a finite number of times, note that, if  $A$  is a clause of a formula, then applications of any of the rules yields a formula which contains the clause  $B$  (possibly  $A$  itself) which is subsumed by  $A$ . It follows, because of proviso (b), that rule (3) cannot "yield" the same clause  $C$  twice. Since there are only a finite number of (distinct) clauses which are implicants of a given formula, it follows that rule (3) can be applied only a finite number of times. Since applications of rules (1) and (2) reduce the number of occurrences of literals in the formula and since rule (3) can be applied only a finite number of times, it follows that rules (1) and (2) can be applied only a finite number of times.

It will now be shown that if the prime implicant  $A$  of a formula  $G$  (not a tautology) is not among the clauses of  $G$ , then either rule (2) or (3) is applicable to  $G$ . Since  $G$  is not a tautology, there is an assignment of truth values to the letters occurring in  $G$  which will make  $G$  false. Since  $(A \rightarrow G)$  is a tautology, it cannot be that no letter in  $A$  occurs in  $G$ , for if that were the case, an assignment of truth values to the letters of  $A$  could be made (independent of the assignment to the letters of  $G$ ) which would make  $A$  true. If  $A$  is of the form  $q_i B$  (where  $q_i$  is  $p_i$  or  $\bar{p}_i$ ) and  $p_i$  does not occur in  $G$ , then  $(\bar{t}B \rightarrow G)$  is a tautology so that  $(B \rightarrow G)$  is a tautology, which contradicts the assumption that  $A$  is prime. Hence each letter which occurs in  $A$  occurs in  $G$ . Furthermore  $A$  subsumes no clause of  $G$ , for each such clause is an implicant of  $G$  while  $A$  is a prime implicant. Thus  $A$  has the properties: (a) it subsumes  $A$ ; (b) it subsumes no clause of  $G$ ; (c) every letter which occurs in  $A$  occurs in  $G$ . There is a longest conjunction  $L$  (with no duplications of letters) which satisfies (a), (b), and (c). Note that, since  $L$  subsumes  $A$ ,  $L$  is an implicant of  $A$ , and since  $A$  is an implicant of  $G$ , it follows that  $L$  is an implicant of  $G$ . It cannot be that every letter in  $G$  occurs in  $L$ , for if this were the case, then for every clause  $C$  of  $G$ , every letter in  $C$  occurs in  $L$  but since  $L$  does not subsume  $C$ , by (b), the truth assignment to the letters of  $L$  which makes  $L$  true will make  $C$  false. From this follows  $(L \rightarrow G)$  is not a tautology, which contradicts the fact that  $L$  is an implicant of  $G$ . Thus some letter, say  $p_i$ , occurs in  $G$  but not in  $L$ . Consider  $p_i L$  and  $\bar{p}_i L$ . These conjunctions satisfy (a) and (c). Since  $L$  is a longest formula satisfying (a), (b), and (c), it follows that  $p_i L$  subsumes a clause of  $G$ ;  $p_i$  must occur in this conjunction since  $L$  does not subsume it; call this conjunction  $p_i D$  ( $D$  possibly empty). Similarly  $\bar{p}_i L$  subsumes  $\bar{p}_i E$ . (The con-

junctions  $p_i D$  and  $\bar{p}_i E$  are distinct since both  $p_i$  and  $\bar{p}_i$  cannot occur in the same formula.) Not both  $D$  and  $E$  can be empty for this would mean  $G$  is a tautology. If exactly one of  $D, E$  is empty, then rule (2) is applicable; if neither is empty, then rule (3) is applicable. Thus, if rules (2) and (3) are not applicable to a formula  $G$ , then every prime implicant of  $G$  is a clause of  $G$ .

If rule (1) is not applicable to a formula  $G$ , as well as rules (2) and (3), then every clause of  $G$  is a prime implicant of  $G$ . For suppose the clause  $A$ , of  $G$ , is not a prime implicant. Then since  $A$  is an implicant of  $G$ , it subsumes a prime implicant  $B$  of  $G$ . Since rules (2) and (3) are not applicable (by the argument of the preceding paragraph),  $B$  is a clause of  $G$ , so that rule (1) would be applicable.

To complete the proof of the theorem, it is necessary only to observe that application of any of the rules yields a formula equivalent to the given one [cf. 5.9 (14), (15), (16)].

5.13.2. To recapitulate, suppose a formula  $F$  is given. Applying any of the three rules to  $F$  yields a formula  $F_1$  yields  $F_2$ , etc. The formulas  $F_1, F_2, \dots$  are obtained such that  $F_1 \text{ eq } F_2 \text{ eq } F_3 \text{ eq } \dots$ . The first paragraph of the proof shows that this sequence of formulas must terminate, i.e., there is a formula  $F_m$  in the sequence such that none of the rules is applicable to  $F_m$ . The second paragraph establishes that, if rules (2) and (3) are not applicable to  $F_i$ , then every prime implicant of  $F_i$  (the prime implicants of  $F$  and  $F_i$  are the same since  $F \text{ eq } F_i$ ) is a clause of  $F_i$ . In particular, every prime implicant of  $F_m(F)$  is a clause of  $F_m$ . In addition, by the third paragraph, every clause of  $F_m$  is a prime implicant of  $F_m(F)$ .

5.13.3. Exercises. (1) Prove the following corollary to 5.13.1: if  $F$  is a disjunction of conjunctions of propositional variables, then every prime implicant of  $F$  is a clause of  $F$ . Hence applications of rule (1) above are enough to produce the  $F'$  of the theorem.

(2) Let rule (3') be the same as rule (3) except that the restriction "A and B are both nonempty" is removed. Then the statement obtained from theorem 5.13.1 by replacing rules (2) and (3) by rule (3') is a theorem. If one applies rule (3') successively, without intervening applications of rule (1), one obtains a formula such that every prime implicant of the formula is a clause of the formula. Having obtained such a formula rule (3') is no longer applicable [even after one or more applications of rule (1)] so that  $F'$  (of the theorem) may be obtained from  $F$  by repeated applications of (3') followed by repeated applications of rule (1).

(3) Show that, if  $A$  is an implicant of  $B$  and  $B$  an implicant of  $C$ , then  $A$  is an implicant of  $C$ . [Cf. 5.9 (9).]

(4) Determine whether  $((A \rightarrow B) \rightarrow C) \text{ eq } (A \rightarrow (B \rightarrow C))$ .

5.13.4. The significance of prime implicants for the simplification of formulas (which are disjunctions of conjunctions of literals) is this: if a clause of a formula is not a prime implicant, then it may be replaced by a prime implicant which it subsumes, yielding an equivalent formula which is "simpler," e.g., from the point of view of number of occurrences of propositional variables. (Recall that the number of relay contacts in the associated relay contact network is equal to the number of occurrences of propositional variables.)

5.13.5. If one makes use of 5.13.1 to simplify a formula, it is better strategy to apply rules (1) and (2) as often as possible [rather than to proceed as in 5.13.3 (2), for example]; that is, at any particular time, if (1) or (2) is applicable, one applies either independent of whether (3) is applicable or not, so that (3) is applied only when rules (1) and (2) are not applicable. The method will be illustrated by the following example:

$$(1) \quad \bar{p}_1\bar{p}_2p_3 \vee p_2\bar{p}_3 \vee \bar{p}_1p_4p_5 \vee \bar{p}_2\bar{p}_4 \vee p_3p_4 \vee \bar{p}_5 \vee \bar{p}_1\bar{p}_3p_4 \quad .$$

$$\text{Rule (2) yields: } \bar{p}_1\bar{p}_2p_3 \vee p_2\bar{p}_3 \vee \bar{p}_1p_4 \vee \bar{p}_2\bar{p}_4 \vee p_3p_4 \vee \bar{p}_5 \vee \bar{p}_1\bar{p}_3p_4 \quad .$$

The last clause subsumes the third and may therefore [rule (1)] be deleted. Applying rule (3) to the third and fourth clauses yields the clause  $\bar{p}_1\bar{p}_2$  which is subsumed by the first clause. Thus, the following formula is obtained:

$$(2) \quad p_2\bar{p}_3 \vee \bar{p}_1\bar{p}_4 \vee \bar{p}_2\bar{p}_4 \vee \bar{p}_1\bar{p}_2 \vee p_3p_4 \vee \bar{p}_5 \quad .$$

Applying now rule (3) to first and third clauses, to first and fourth clauses, to first and fifth clauses, and to third and fifth clauses yields:

$$(3) \quad p_2\bar{p}_3 \vee \bar{p}_1p_4 \vee \bar{p}_2\bar{p}_4 \vee \bar{p}_1\bar{p}_2 \vee p_3p_4 \vee \bar{p}_5 \vee [\bar{p}_3\bar{p}_4 \vee \bar{p}_1\bar{p}_3 \vee p_2p_4 \vee \bar{p}_2p_3] \quad .$$

None of the rules is now applicable, so that all prime implicants of the given formula have been obtained. The clauses in brackets are clearly jointly dispensable. Other clauses may be dispensable possibly conditional upon the presence of some or all of the bracketed clauses. If A is the formula obtained from (3) by deleting the first clause, then one may check that  $(p_2\bar{p}_3 \rightarrow A)$  is a tautology, so that  $A \text{ eq } B$  where B is the given formula; however,  $p_2\bar{p}_3$  is not dispensable from (2). Notice that  $p_1p_2$  is dispensable from (2) since it was obtained by rule (3) applied to  $\bar{p}_1\bar{p}_2$  is dispensable from (2) since it was obtained by rule (3) applied to  $\bar{p}_1p_4$  and  $\bar{p}_2p_4$ , which are present in (2). Another way of seeing that  $\bar{p}_1\bar{p}_2$  is dispensable is to note that  $(\bar{p}_1\bar{p}_2 \rightarrow C)$  is a tautology where C is (2) with  $\bar{p}_1\bar{p}_2$  deleted. Thus the given formula is equivalent to

$$(4) \quad p_2\bar{p}_3 \vee \bar{p}_1p_4 \vee \bar{p}_2\bar{p}_4 \vee p_3p_4 \vee \bar{p}_5 \vee [\bar{p}_1\bar{p}_2 \vee \bar{p}_3\bar{p}_4 \vee \bar{p}_1\bar{p}_3 \vee p_2p_4 \vee \bar{p}_2p_3] \quad ,$$

where the clauses in brackets are simultaneously dispensable. To check whether this formula is "simplest" would require much tedious labor. We will not carry the argument further.

As the argument has, perhaps, already demonstrated, the simplification pro-

cedure is straightforward and reasonably quick up to the point where one obtains all prime implicants. From that point on the procedure is in complicated cases quite tedious and lengthy if one wants to be sure he has a simplest equivalent of the given formula. Notice, however, that the given formula has sixteen occurrences of literals while our simplest equivalent has only nine.

For further examples and simplification suggestions the reader is referred to item 2 of the bibliography, page 1.

5.14. A law of duality. The dual  $A^D$  of a well-formed formula  $A$  is now defined inductively on the length of a formula.

- (1) If  $A$  is  $t(f)$ , then  $A^D$  is  $f(t)$ ; if  $A$  is  $p_i$ , then  $A^D$  is  $p_i$ .
- (2) If  $A$  is  $(B \vee C)$ , then  $A^D$  is  $B^D \wedge C^D$ ; if  $A$  is  $(B \wedge C)$ , then  $A^D$  is  $B^D \vee C^D$ .
- (3) If  $A$  is  $\sim B$ , then  $A^D$  is  $\sim(B^D)$ .

Thus, for example, if  $A$  is  $p_1 \vee (\sim(p_2 \wedge p_3))$ , then  $A^D$  is  $p_1 \wedge (\sim(p_2 \vee p_3))$ .

Lemma. If  $g$  is the function (of  $n$  variables) associated with the formula  $A$  (cf. 5.3) and  $g^D$  is the function associated with  $A^D$ , then  $g^D(x_1, x_2, \dots, x_n) = \overline{g(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$ , where the bar over  $g$  indicates the complement of  $g(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)$ ,  $\overline{t}$  is  $f$ , and  $\overline{f}$  is  $t$ .

The proof is by induction on the length of the well-formed formula.

Proof. If  $A$  is  $t$ , then  $A^D$  is  $f$ ,  $g(x_1, x_2, \dots, x_n) = t$ ,  $g^D(x_1, x_2, \dots, x_n) = f$ ,  $g(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n) = t$ , and  $\overline{g(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)} = f$ , so that the conclusion of the lemma is satisfied. If  $A$  is  $f$ , the argument is analogous. If  $A$  is  $p_i$ , then  $A^D$  is  $p_i$ ,  $g(x_1, x_2, \dots, x_n) = x_i$ ,  $g^D(x_1, x_2, \dots, x_n) = x_i$ ,  $g(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n) = \overline{x}_i$ , and  $\overline{g(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)} = x_i$ , so that the conclusion of the lemma is satisfied in this case. Now suppose  $A$  is  $B \vee C$ ,  $g_B^D(x_1, x_2, \dots, x_n) = \overline{g_B(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$ , and  $g_C^D(x_1, x_2, \dots, x_n) = \overline{g_C(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$ . Then  $g_A(x_1, x_2, \dots, x_n) = g_B(x_1, x_2, \dots, x_n) \cup g_C(x_1, x_2, \dots, x_n)$ , so that

$$\begin{aligned} \overline{g_A(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)} &= \overline{g_B(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n) \cup g_C(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)} \\ &= \overline{g_B(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)} \cap \overline{g_C(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)} \\ &= g_B^D(x_1, x_2, \dots, x_n) \cap g_C^D(x_1, x_2, \dots, x_n) \end{aligned}$$

Since  $A^D$  is  $B^D \wedge C^D$ , it follows that  $g_A^D(x_1, x_2, \dots, x_n) = g_B^D(x_1, x_2, \dots, x_n) \cap g_C^D(x_1, x_2, \dots, x_n)$ , so that  $g_A^D(x_1, x_2, \dots, x_n) = \overline{g_A(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$ . The argument in the case  $A$  is  $B \wedge C$  is analogous.

Now suppose  $A$  is  $\sim B$  and  $g_B^D(x_1, x_2, \dots, x_n) = \overline{g_B(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$ . Then  $g_A(x_1, x_2, \dots, x_n) = \overline{g_B(x_1, x_2, \dots, x_n)}$ , so that  $\overline{g_A(x_1, x_2, \dots, x_n)} = g_B(x_1, x_2, \dots, x_n) = \overline{g_B(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$ . Since

$A^D$  is  $\sim B^D$ , it follows that  $g_A^D(x_1, x_2, \dots, x_n) = \overline{g_B^D(x_1, x_2, \dots, x_n)}$ , so that (using inductive assumption)  $g_A^D(x_1, x_2, \dots, x_n) = \overline{g_B(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})} = g_B(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}) = \overline{g_A(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})}$ .

Theorem. If  $A \text{ eq } B$ , then  $A^D \text{ eq } B^D$ .

Proof. The result is immediate from the lemma:  $g_A^D(x_1, \dots, x_n) = \overline{g_A(\overline{x_1}, \dots, \overline{x_n})} = \overline{g_B(\overline{x_1}, \dots, \overline{x_n})} = g_B^D(x_1, \dots, x_n)$ .

Corollary 1. If  $A$  is a tautology, then  $\sim(A^D)$  is a tautology.

Proof. Since  $A \text{ eq } t$ , it follows that  $A^D \text{ eq } t^D$ ,  $A^D \text{ eq } f$ , and  $\sim(A^D) \text{ eq } t$ .

Corollary 2. If  $(A \equiv B)$  is a tautology (cf. 5.9), then  $(A^D \equiv B^D)$  is a tautology.

Proof. By previous corollary  $\sim((A \equiv B)^D)$  is a tautology. On the other hand  $(A \equiv B)^D \text{ eq } \sim(A^D \equiv B^D)$  (see exercise 1 below), so that  $(A^D \equiv B^D)$  is a tautology.

Exercises. Show (cf. 5.9) that

- (1)  $(A \equiv B)^D \text{ eq } \sim(A^D \equiv B^D)$ ;
- (2)  $\sim(A \equiv B) \text{ eq } (\sim A) \equiv B \text{ eq } A \equiv (\sim B) \text{ eq } A + B$ ;
- (3)  $(A \uparrow B)^D \text{ eq } (A^D \downarrow B^D)$ ;
- (4)  $(A \rightarrow B)^D \text{ eq } \sim(B^D \rightarrow A^D)$ ;
- (5) if  $A \rightarrow B$  is a tautology then  $(B^D \rightarrow A^D)$  is a tautology;
- (6) if  $\text{Eq}(A) \leq \text{Eq}(B)$ , then  $\text{Eq}(B^D) \leq \text{Eq}(A^D)$ ;
- (7)  $A \equiv B \equiv C \text{ eq } A + B + C$ .
- (8) Every formula is equivalent to one in "expanded conjunctive normal form." Define this form by analogy with 5.12 (a). Use the result of 5.12 (a) and a duality law to obtain the desired result.

5.15. Preliminary to establishing a representation theorem for finite Boolean algebras, we note some further properties of Boolean algebras.

- 5.15.1. (1)  $x \leq x \vee y$ ;  $x \geq x \wedge y$ ;
- (2) if  $x \leq y$  and  $x' \leq y'$ , then  $x \vee x' \leq y \vee y'$  and  $x \wedge x' \leq y \wedge y'$ ;
- (3) if  $x \leq y$  and  $x \leq y'$ , then  $x \leq y \wedge y'$ , and if  $x \geq y$  and  $x \geq y'$ , then  $x \geq y \vee y'$  [these statements are immediate consequences of (2)];
- (4)  $x \leq y$  if and only if  $x \wedge \overline{y} = 0$  if and only if  $\overline{x} \vee y = 1$ .

Proof. Suppose  $x \leq y$ . Then  $x \wedge y = x$  and  $0 = (x \wedge y) \wedge \overline{y} = x \wedge \overline{y}$ . Suppose  $x \wedge \overline{y} = 0$ . Then  $x \vee \overline{y} = (x \wedge y) \vee y = y$ , so that  $x \leq y$ .

- (5) An element  $a$  in a Boolean algebra is minimal if and only if  $a \neq 0$ , and for all  $x$  if  $x \leq a$ , then  $x = a$  or  $x = 0$ . In a finite Boolean algebra, given any element  $y \neq 0$ , there exists a minimal element less than or equal to  $y$ .

- (6) An element  $a$  in a Boolean algebra is an atom if and only if  $a \neq 0$  and for all  $x$  either  $x \wedge a = a$  or  $x \wedge a = 0$ . An element is minimal if and only if it is an atom.
- (7) If  $a, b$  are atoms and  $ab \neq 0$  then  $a = b$ .
- (8) Let  $a, a_1, a_2, \dots, a_n$  be atoms of an arbitrary Boolean algebra.
- If  $x = a_1 \vee a_2 \vee \dots \vee a_n$  and  $a \leq x$ , then  $a = a_i$  for some  $i$ .
  - $a \leq y \vee z$  if and only if  $a \leq y$  or  $a \leq z$ .
  - $a \leq y \wedge z$  if and only if  $a \leq y$  and  $a \leq z$ .

Exercise. Prove (1), (2), (3), (5), (6), (7), and (8) above.

5.15.2. Lemma. In a finite Boolean algebra, every element  $x$  is the join (least upper bound) of all atoms  $a$  in the algebra such that  $a \leq x$ .

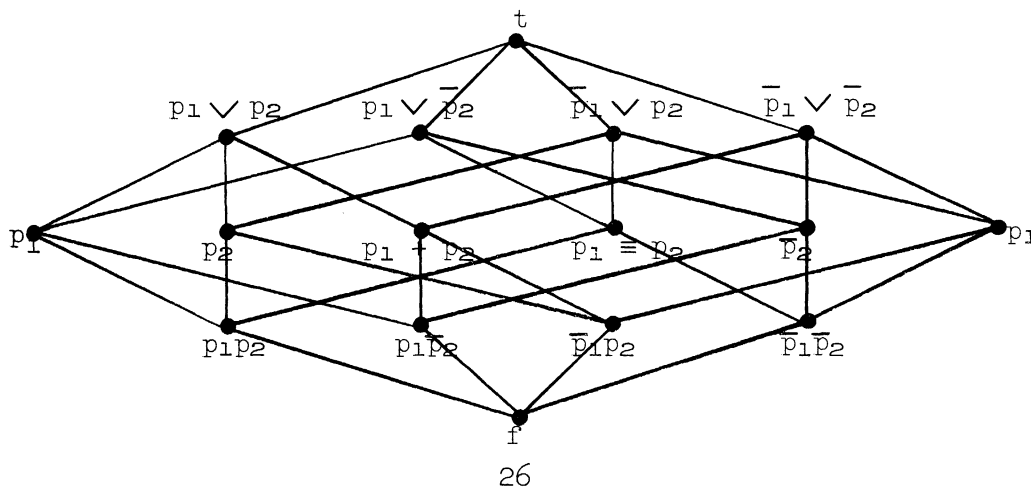
Proof. Suppose  $a_1, a_2, \dots, a_n$  are all the atoms less than or equal to  $x$ . Then the element  $y = (a_1 \vee a_2 \vee \dots \vee a_n) \leq x$  [cf. 5.15.1 (3)]. It is sufficient then to show that  $x \leq y$  which is equivalent [cf. 5.15.1 (4)] to showing  $x\bar{y} = 0$ . We shall show that the assumption that  $x\bar{y} \neq 0$  leads to a contradiction from which the desired result follows. If  $x\bar{y} \neq 0$ , then there is an atom [5.15.1 (5)]  $a \leq x\bar{y}$  so that  $a \leq x$  and  $a \leq \bar{y}$ . From  $a \leq x$  and the choice of the  $a_i$ 's it follows that  $a = a_i$  for some  $i$ . Hence  $a \leq y$ . Thus  $a \leq y \wedge \bar{y} = 0$  [cf. 5.15.1 (3)], so that  $a = 0$ , contradicting the choice of  $a$ .

5.15.3. Theorem. If  $(A, \vee, \wedge)$  is a Boolean algebra, it is isomorphic to  $(C, \cup, \cap)$  where  $C$  is the class of all sets of atoms of  $A$ . Indeed, the function  $h$  which associates with each  $x \in A$  the set  $h(x)$  of atoms  $a$  in  $A$  such that  $a \leq x$  is an isomorphism of  $(A, \vee, \wedge)$  onto  $(C, \cup, \cap)$ .

Proof. Immediate from 5.15.2 and 5.15.1 (8). (Cf. 4.19.)

Corollary. If  $r$  is the number of elements in a finite Boolean algebra, then  $r = 2^m$  for some  $m$ . Two finite Boolean algebras are isomorphic if and only if they have the same number of elements.

5.15.4. Example. Below is a lattice diagram of the Boolean algebra associated with the formulas of the propositional calculus in the case  $n = 2$ . (Cf.



5.1 and 5.6.) Recall that  $p_1 + p_2$  is an abbreviation for  $p_1\bar{p}_2 \vee \bar{p}_1p_2$  and  $p_1 \equiv p_2$  is an abbreviation for  $p_1p_2 \vee \bar{p}_1\bar{p}_2$ .

The atoms of this algebra are:  $p_1p_2, p_1\bar{p}_2, \bar{p}_1p_2, \bar{p}_1\bar{p}_2$ . The isomorphism which the theorem establishes applied to this case is:

$$\begin{array}{ll}
 h(f) = \emptyset & h(p_1 \equiv p_2) = \{p_1p_2, \bar{p}_1\bar{p}_2\} \\
 h(p_1p_2) = \{p_1p_2\} & h(\bar{p}_2) = \{p_1\bar{p}_2, \bar{p}_1\bar{p}_2\} \\
 h(p_1\bar{p}_2) = \{p_1\bar{p}_2\} & h(\bar{p}_1) = \{\bar{p}_1p_2, \bar{p}_1\bar{p}_2\} \\
 h(\bar{p}_1p_2) = \{\bar{p}_1p_2\} & h(p_1 \vee p_2) = \{p_1p_2, p_1\bar{p}_2, \bar{p}_1p_2\} \\
 h(\bar{p}_1\bar{p}_2) = \{\bar{p}_1\bar{p}_2\} & h(p_1 \vee \bar{p}_2) = \{p_1p_2, p_1\bar{p}_2, \bar{p}_1\bar{p}_2\} \\
 h(p_1) = \{p_1p_2, p_1\bar{p}_2\} & h(\bar{p}_1 \vee \bar{p}_2) = \{\bar{p}_1p_2, \bar{p}_1\bar{p}_2, \bar{p}_1\bar{p}_2\} \\
 h(p_2) = \{p_1p_2, \bar{p}_1p_2\} & h(\bar{p}_1 \vee p_2) = \{p_1\bar{p}_2, \bar{p}_1p_2, \bar{p}_1\bar{p}_2\} \\
 h(p_1+p_2) = \{p_1\bar{p}_2, \bar{p}_1p_2\} & h(t) = \{p_1p_2, p_1\bar{p}_2, \bar{p}_1p_2, \bar{p}_1\bar{p}_2\}
 \end{array}$$

Notice that 5.15.2 provides the information that  $x$  is the join of the elements in  $h(x)$ , e.g.,  $p_1 \vee \bar{p}_2 = p_1p_2 \vee p_1\bar{p}_2 \vee \bar{p}_1p_2$ ; the right-hand member of the equality is the expanded disjunctive normal form of  $p_1 \vee \bar{p}_2$ .

5.15.5. Exercises. (1) In a manner analogous to the correspondence established on page 10, establish a correspondence between 4-tuples of 0's and 1's and the set of all subsets of  $p_1p_2, p_1\bar{p}_2, \bar{p}_1p_2, \bar{p}_1\bar{p}_2$ . This correspondence together with the correspondence  $h$  in the example immediately above yield a correspondence between the elements of the Boolean algebra depicted in the lattice diagram and 4-tuples of 0's and 1's. This correspondence is an isomorphism. Relabel the lattice diagram of the example above in a way indicated by this correspondence and check that join and meet are preserved under this correspondence. Note that such a correspondence may also be obtained by associating with a formula the sequence of values of its associated function, for a fixed ordering of its arguments. (Cf. 5.3.)

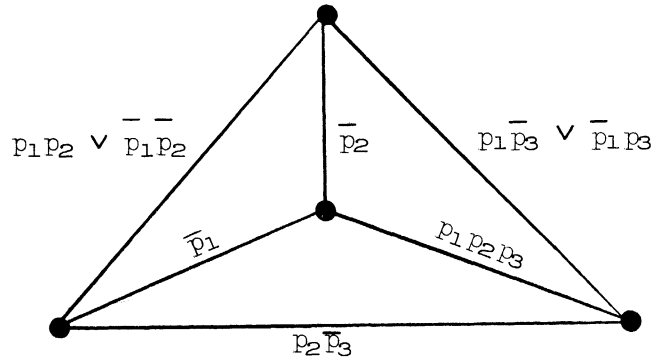
(2) A set  $S$  of elements is closed with respect to a binary ( $n$ -ary) operation  $\circ$  if and only if  $x_1 \circ x_2$  [ $\circ x_1, x_2 \dots x_n$ ] is in  $S$  whenever  $x_1, x_2$  [ $x_1, x_2, \dots, x_n$ ] is in  $S$ . What is the smallest class of elements in the 16-element algebra depicted above containing

- (a)  $p_1, p_1+p_2$ , and closed under join and meet?
- (b)  $p_1, p_1+p_2$ , and closed under join, meet, and complement?
- (c)  $p_1 \equiv p_2, p_1+p_2$ , and closed under join, meet, and complement?

(3) An isomorphism of an algebra onto itself is called an automorphism. Show that a Boolean algebra with  $2^n$  elements has exactly  $n!$  automorphisms. Hint: In any automorphism, atoms will go into (be associated with) atoms.

6. MULTI-TERMINAL NETWORKS<sup>39,27</sup>

6.1. An example.



Consider a particular truth assignment, say 1, 0, 0 (i.e., t, f, f) to  $p_1, p_2, p_3$ , respectively. Then the following matrix

$$\begin{array}{c}
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{cccc}
 & 1 & 2 & 3 & 4 \\
 1 & \left( \begin{array}{cccc}
 1 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 \\
 0 & 1 & 1 & 0 \\
 0 & 1 & 0 & 1
 \end{array} \right) \\
 2 \\
 3 \\
 4
 \end{array}$$

has a "1" in the  $i, j$  position (i.e., the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column counting from top to bottom and left to right, respectively) if there is a closed path between node  $i$  and node  $j$  (for this truth assignment); otherwise the  $i, j^{\text{th}}$  entry is "0." The convention that there is a closed path between a node and itself is adopted. Let the  $i, j^{\text{th}}$  entry be  $a_{ij}$ . Then the matrix can also be described as follows:

$$a_{ii} = 1, \quad a_{ij} = a_{ji}, \quad a_{23} = a_{24} = 1, \quad \text{and} \quad a_{12} = a_{13} = a_{14} = a_{34} = 0.$$

6.2. Pierce and matrix product. With each binary relation  $R$  defined over a finite set  $S$  of  $n$  elements,  $n \geq 1$ , a matrix  $M(R)$  of 0's and 1's is associated as follows. We assume  $S = \{1, 2, \dots, n\}$ . The  $i, j^{\text{th}}$  entry of  $M(R)$  is 1 if  $iRj$ , otherwise 0. Every  $n \times n$  matrix ( $n$  rows and  $n$  columns) of 0's and 1's is  $M(R)$  for some  $R$ . For example, the matrix of 6.1 is  $M(R)$  if  $R$  is reflexive over 1, 2, 3, 4, symmetric, and  $2R3, 2R4, 1R2, 1R3, 1R4, 3R4$ .



The (Pierce) product  $R_1 \times R_2$  of two binary relations is defined as follows:  $x(R_1 \times R_2)y$  if and only if there exists an element  $z$  such that  $xR_1z$  and  $zR_2y$ . The (matrix) product  $C = A \times B$  of two  $n \times n$  matrices  $A = (a_{ij})$  and  $B = (b_{ij})$  is defined as follows:  $c_{ij} = \sum_{k=1}^n a_{ik} \circ b_{kj}$ . (It is convenient to use the same symbol "x" for both Pierce and matrix product.) This makes sense if an operation "+" and an operation "." are defined over the elements denoted by the entries of the matrix. (It is also assumed here that "+" is associative.) We shall be interested in the case where the matrix entries denote elements of a Boolean algebra (more generally, a lattice) and will, therefore, write

$$c_{ij} = \bigvee_{k=1}^n a_{ik} \wedge b_{kj}$$

and " $\wedge$ " will sometimes be omitted (but understood). This matrix product will sometimes be referred to as "the Boolean matrix product." Notice that, if A and B have 1's along the main diagonal then so will C, since  $c_{ii} = \bigvee_{k=1}^n a_{ik} b_{ki} = 1$  since among these disjuncts is  $a_{ii} b_{ii} = 1$ .

6.2.1. Theorem. If  $R_1, R_2$  are binary relations defined over the same set S of n elements (assumed without loss of generality to be the first n positive integers), then

$$M(R_1 \times R_2) = M(R_1) \times M(R_2) .$$

Proof. Let  $c_{ij}, a_{ij}, b_{ij}$  be the  $i, j^{\text{th}}$  element of the respective matrices  $M(R_1 \times R_2), M(R_1), M(R_2)$ . Then

$$c_{ij} = \bigvee_{k=1}^n (a_{ik} \wedge b_{kj})$$

so that  $c_{ij} = 1$  if and only if there exists an integer  $k, 1 \leq k \leq n$ , such that  $a_{ik} = 1 = b_{kj}$  if and only if there exists  $k \in S$  such that  $iR_1k$  and  $kR_2j$  if and only if  $i(R_1 \times R_2)j$ .

6.2.2. Corollary. The function M is an isomorphism between the system of all binary relations (defined over a fixed set of n elements) under Pierce product,  $\vee, \wedge$  and the system consisting of the set of all  $n \times n$  matrices with 0, 1 entries, under Boolean matrix product and  $\vee, \wedge$ .

Let  $A = (a_{ij}), B = (b_{ij})$  be  $n \times n$  matrices with all the  $a_{ij}, b_{ij}$  elements of some lattice. The  $i, j^{\text{th}}$  element of  $A \vee B$  is by definition  $a_{ij} \vee b_{ij}$ .  $A \wedge B$  is defined analogously. For binary relations  $x(R_1 \vee R_2)y$  if and only if  $xR_1y$  or  $xR_2y$ .  $R_1 \wedge R_2$  is defined similarly.

6.2.3. Theorem.

- (a) The set of  $n \times n$  matrices of 6.2.2 form a lattice under  $\vee, \wedge$ , and if  $L$  is a Boolean algebra, then this lattice is a Boolean algebra.
- (b) The subset of matrices of (a) with 1's along the main diagonal (i.e.,  $a_{ii} = 1$  for  $1 \leq i \leq n$ ) is a Boolean algebra, if  $L$  is a Boolean algebra.
- (c)  $A \times (B \vee C) = (A \times B) \vee (A \times C)$ .
- (d)  $(A \times B) \times C = A \times (B \times C)$ .

Proof.

- (a) It is obvious that this system of matrices forms a lattice, say  $\mathcal{L}$ .  $\mathcal{L}$  has a zero (one) element, namely, the matrix all of whose entries are "0" ("1"). If  $A = (a_{ij})$ , it is easily seen that  $\bar{A} = (\bar{a}_{ij})$ .
- (b) It is obvious that this system of matrices forms a lattice, say  $\mathcal{L}'$ .  $\mathcal{L}'$  has a zero element, namely, the matrix all of whose entries are "0" except the elements along the main diagonal which are all "1."  $\mathcal{L}'$  has a one element, namely, the matrix all of whose entries are "1." If  $A = (a_{ij})$ ,  $B = (b_{ij})$ , and  $b_{ij} = \bar{a}_{ij}$  for  $i \neq j$  and  $b_{ii} = 1$ , then  $B = \bar{A}$ .
- (c) The  $i, j^{\text{th}}$  element of  $A \times (B \vee C)$  is

$$\bigvee_{k=1}^n a_{ik} (b_{kj} \vee c_{kj}) = \bigvee_{k=1}^n (a_{ik} b_{kj} \vee a_{ik} c_{kj}) =$$

$$\left( \bigvee_{k=1}^n a_{ik} b_{kj} \right) \vee \left( \bigvee_{k=1}^n a_{ik} c_{kj} \right)$$

which is the  $i, j^{\text{th}}$  element of  $A \times B$  and  $A \times C$ .

- (d) The  $i, j^{\text{th}}$  element of  $(A \times B) \times C$  is

$$\bigvee_{k=1}^n \left( \bigvee_{\ell=1}^n a_{i\ell} b_{\ell k} \right) c_{kj} = \bigvee_{\ell=1}^n a_{i\ell} b_{\ell k} c_{kj} = \bigvee_{\ell=1}^n a_{i\ell} \left( \bigvee_{k=1}^n b_{\ell k} c_{kj} \right)$$

which is the  $i, j^{\text{th}}$  element of  $A \times (B \times C)$ .

6.3. Connection matrix. An  $n \times n$  symmetric matrix with 1's along the main diagonal and with entries which are formulas of the propositional calculus may be associated with those relay contact networks constructed as follows: With each  $i, j$ ,  $i < j$ , a two-terminal network with terminals  $i, j$  is constructed whose behavior is given by the propositional formula which is the  $i, j^{\text{th}}$  entry of the given matrix. The two-terminal networks are then "connected" together by con-

necting (identifying) those terminals with the same name. If each two-terminal network with terminals  $i, j$  is so chosen to be the series-parallel network whose "structure" (cf. 5.11) as well as "behavior" is given by the  $i, j^{\text{th}}$  entry of the matrix, then the (labeled) network associated with the matrix is "essentially" unique. Conversely, given a network, if one "labels" enough of its nodes,  $1, 2, \dots, n$ , there is an  $n \times n$  matrix (of the kind mentioned above) which has the given network as an associated (in the above-mentioned way) network. By choosing enough nodes, the two-terminal network associated with the  $i, j^{\text{th}}$  entry of the correlated matrix will be series-parallel. A matrix correlated with a network in this way is called a connection matrix of the network. Any symmetric matrix with 1's on the main diagonal and propositional formulas as entries is a connection matrix of some labeled network. A labeled network corresponding to the diagram of 6.1 has as its connection matrix the symmetric matrix  $A = (a_{ij})$  with 1's along the main diagonal and

$$\begin{aligned} a_{12} &= p_1 p_2 \vee \bar{p}_1 \bar{p}_2, & a_{13} &= p_2 \bar{p}_3, & a_{14} &= \bar{p}_1, \\ a_{23} &= p_1 \bar{p}_3 \vee \bar{p}_1 p_3, & a_{24} &= \bar{p}_2, & a_{34} &= p_1 p_2 p_3. \end{aligned}$$

6.4. Characteristic matrix. The characteristic matrix  $\chi(A)$  of a labeled network with connection matrix  $A$  is a matrix with the property: the  $i, j^{\text{th}}$  entry is a propositional formula which takes on the value 1 for those and only those truth assignments which correspond (cf. 5.11) to states of the relay coils which result in there being some closed path between nodes  $i$  and  $j$ .

In example 6.1 the characteristic matrix  $\chi(A) = (b_{ij})$  where

$$\begin{aligned} b_{12} &= p_1 p_2 \vee \bar{p}_1 \bar{p}_2, & b_{13} &= p_2 \bar{p}_3 \vee \bar{p}_1 \bar{p}_2 p_3, & b_{14} &= \bar{p}_1, \\ b_{23} &= p_1 \bar{p}_3 \vee \bar{p}_1 p_3, & b_{24} &= \bar{p}_2, & b_{34} &= p_1 p_3 \vee \bar{p}_1 p_2 \bar{p}_3 \vee \bar{p}_1 \bar{p}_2 p_3. \end{aligned}$$

We indicate the calculation for  $b_{13}$ . The condition under which there is a closed path from node 1 to node 3 and not passing through any labeled nodes is given by  $p_2 \bar{p}_3$ . The conditions under which there is a closed path from node 1 to node 2 and from node 2 to node 3 and not passing through any labeled nodes (except 2) is given by  $(p_1 p_2 \vee \bar{p}_1 \bar{p}_2)(p_1 \bar{p}_3 \vee \bar{p}_1 p_3)$ . Similarly "corresponding" to  $143$  is  $\bar{p}_1(p_1 p_2 p_3)$ , to  $1243$  is  $(p_1 p_2 \vee \bar{p}_1 \bar{p}_2) \bar{p}_2 (p_1 p_2 p_3)$ , and to  $1423$  is  $\bar{p}_1(\bar{p}_2)(p_1 \bar{p}_3 \vee \bar{p}_1 p_3)$ . Thus the condition under which there is any closed path from node 1 to node 3 is given by  $(p_2 \bar{p}_3) \vee (p_1 p_2 \vee \bar{p}_1 \bar{p}_2)(p_1 \bar{p}_3 \vee \bar{p}_1 p_3) \vee \bar{p}_1(p_1 p_2 p_3) \vee (p_1 p_2 \vee \bar{p}_1 \bar{p}_2) \bar{p}_2 (p_1 p_2 p_3) \vee \bar{p}_1 \bar{p}_2 (p_1 \bar{p}_3 \vee \bar{p}_1 p_3)$  which simplifies to  $p_2 \bar{p}_3 \vee \bar{p}_1 \bar{p}_2 p_3$ .

## 6.5. Analysis theorem.

6.5.1. Theorem. If  $A$  is the  $n \times n$  connection matrix of a labeled network,

then  $A \leq A^2 \leq A^3 \leq \dots \leq A^r = A^{r+1} = \dots = A^{n-1} = \dots = \chi(A)$ , where  $r < n$ .

Proof. Consider a fixed truth assignment  $T$  to the propositional variables occurring in the entries of  $A$  yielding a matrix  $A_T$  of 0's and 1's. Let  $R_T$  be the associated relation, i.e.,  $M(R_T) = A_T$  (cf. 6.2). Then  $iR_T^m j$ ,  $m \geq 1$ , means there is a closed path (under this truth assignment) from node  $i$  to node  $j$  passing through at most  $m-1$  labeled nodes of the network other than  $i, j$ . Since there are only  $n-2$  such nodes, it follows that  $R_T^{n-1} = R_T^n = R_T^{n+1} = \dots$ . Hence, by 6.2.1,  $A_T^{n-1} = A_T^n = A_T^{n+1} = \dots$ . Since  $R_T \leq R_T^2 \leq R_T^3 \leq \dots$  (i.e.,  $xR_T y$  implies  $xR_T^2 y$ , etc., or still otherwise stated,  $R_T \vee R_T^2 = R_T^2$ , etc.), it follows (6.2.2) that  $A_T \leq A_T^2 \leq A_T^3 \leq \dots$ . Inasmuch as these relations hold for each truth assignment  $T$  and  $(A \times B)_T = A_T \times B_T$  (by 5.3 and the definition of matrix product), the results follows.

6.5.2. Corollary.  $A = A^2$  if and only if  $A = \chi(A)$ .

Proof. If  $A = A^2$ , then  $A \times A = A^2 \times A$  so that  $A = A^2 = A^3$ . Similarly  $A = A^{n-1} = \chi(A)$ . If  $A = \chi(A)$  then  $A = A^{n-1}$  and  $A^2 = A^n = \chi(A) = A$ .

6.6. Synthesis theorem.

6.6.1. Lemma. If  $A, B, C, D$  and  $n \times n$  matrices whose entries denote elements of a lattice and if  $A \leq C$  and  $B \leq D$ , then  $A \times B \leq C \times D$ .

Proof. The  $i, j$ <sup>th</sup> elements of  $A \times B$  and  $C \times D$  are, respectively,

$$\bigvee_{k=1}^n a_{ik} b_{kj} \quad , \quad \bigvee_{k=1}^n c_{ik} d_{kj} \quad .$$

Since  $A \leq C$ , it follows  $a_{ik} \leq c_{ik}$ . Similarly  $b_{kj} \leq d_{kj}$ . Hence  $a_{ik} b_{kj} \leq c_{ik} d_{kj}$ . It follows that

$$\bigvee_{k=1}^n a_{ik} b_{kj} \leq \bigvee_{k=1}^n c_{ik} d_{kj} \quad .$$

6.6.2. Corollary. If  $A, B$  are connection matrices then (a)  $A \leq \chi(B)$  if and only if  $\chi(A) \leq \chi(B)$ ; if  $A \leq B$ , then  $\chi(A) \leq \chi(B)$ .

Proof.

(a) If  $\chi(A) \leq \chi(B)$ , then  $A \leq \chi(B)$  since  $A \leq \chi(A)$ . If  $A \leq \chi(B)$ , then  $A \times A \leq \chi(B) \times \chi(B) = \chi(B)$ . Similarly  $A^4 \leq \chi(B)$ , etc. Hence  $\chi(A) \leq \chi(B)$ .

(b) If  $A \leq B$ , then  $A \leq \chi(B)$  and (a) gives the result.

6.6.3. Theorem. If  $\chi(X) \leq \chi(Y) \leq \chi(A)$ , then  $\chi(A) = \chi((A \wedge \bar{X}) \vee Y)$ , where  $\bar{X}$  is understood in the sense of 6.2.3 (b) (i.e.,  $\bar{X}$  is obtained from  $X$  by comple-

menting each element of the main diagonal).

Proof. It will be shown that (a)  $\chi(A) \leq \chi(A\bar{X}\vee Y)$  and (b)  $\chi(A\bar{X}\vee Y) \leq \chi(A)$ . There will be several applications of 6.6.2 (a). Note that since  $X \leq \chi(Y)$ , it follows [5.15.1 (4)] that  $\bar{X}\vee\chi(Y) = 1$  (the all 1 matrix). Hence  $A = A(\bar{X}\vee\chi(Y)) = A(\bar{X}\vee\chi(Y))$  so that  $A \leq \bar{X}\vee\chi(Y)$ . Now  $\bar{X}\vee\chi(Y) \leq (\bar{X}\vee Y)^{n-1} = \chi(\bar{X}\vee Y)$  since among the disjuncts of  $(\bar{X}\vee Y)^{n-1}$ ,  $(\bar{X})^{n-1}$  and  $Y^{n-1}$  appear. Thus,  $A \leq \chi(\bar{X}\vee Y)$  and (a) follows. To obtain (b), note that  $A\bar{X} \leq A \leq \chi(A)$  and  $Y < \chi(A)$  so that  $A\bar{X}\vee Y \leq \chi(A)$  and (b) follows.

6.6.5. Theorem. If  $\chi(A) = \chi(B)$  and if  $A \leq C \leq B$ , then

- (a)  $\chi(A) = \chi(C)$ ,
- (b)  $\chi(A) = \chi(A\vee B)$ ,
- (c)  $\chi(A) = \chi(A \times B)$ .

Proof.

- (a) Immediate from 6.6.2 (b) and hypothesis.
- (b)  $A \leq A\vee B$  so that  $\chi(A) \leq \chi(A\vee B)$ .  
 $A\vee B \leq \chi(A)\vee\chi(B) = \chi(A)$  so that  $\chi(A\vee B) \leq \chi(A)$ .
- (c) Analogous to (b).

6.6.5. The results of 6.6.3 and 6.6.4 are now applied to 6.1 (cf. 6.3, 6.4).

$$A = \begin{pmatrix} 1 & p_1p_2\vee\bar{p}_1\bar{p}_2 & p_2\bar{p}_3 & \bar{p}_1 \\ & 1 & p_1\bar{p}_3\vee p_1p_3 & \bar{p}_2 \\ & & 1 & p_1p_2p_3 \\ & & & 1 \end{pmatrix}$$

$$\chi(A) = \begin{pmatrix} 1 & p_1p_2\vee\bar{p}_1\bar{p}_2 & p_2\bar{p}_3\vee\bar{p}_1\bar{p}_2p_3 & \bar{p}_1 \\ & 1 & p_1\bar{p}_3\vee\bar{p}_1p_3 & \bar{p}_2 \\ & & 1 & b_{34} \\ & & & 1 \end{pmatrix}$$

where  $b_{34} = p_1p_3\vee\bar{p}_1p_2\bar{p}_3\vee\bar{p}_1\bar{p}_2p_3$ .

Let  $y_{12} = p_1p_2$ ,  $y_{13} = p_2\bar{p}_3$ ,  $y_{14} = \bar{p}_1$ ,  $y_{23} = \bar{p}_1p_3$ ,  $y_{24} = \bar{p}_2$ ,  $y_{34} = p_1\bar{p}_2p_3$ . Then  $Y \leq \chi(A)$ . Choose  $X = Y^2$  so that  $\chi(X) = \chi(Y)$ . A calculation yields:  
 $x_{12} = p_1p_2\vee\bar{p}_1\bar{p}_2$ ,  $x_{13} = p_2\bar{p}_3$ ,  $x_{14} = \bar{p}_1$ ,  $x_{23} = (\bar{p}_1\vee\bar{p}_2)p_3\vee p_1p_2\bar{p}_3$ ,  $x_{24} = \bar{p}_2$ ,  
 $x_{34} = p_1\bar{p}_2p_3\vee\bar{p}_1p_2\bar{p}_3\vee\bar{p}_1\bar{p}_2p_3$ . If  $Z = A\bar{X}\vee Y$ , then a calculation yields:  $z_{12} = p_1p_2$ ,  $z_{13} = p_2\bar{p}_3$ ,  $z_{14} = \bar{p}_1$ ,  $z_{23} = p_1\bar{p}_2\bar{p}_3\vee\bar{p}_1p_3$ ,  $z_{24} = \bar{p}_2$ , and  $z_{34} = p_1p_3$ . Thus  $\chi(Z) = \chi(A)$ . Let  $u_{12} = p_1p_2$ ,  $u_{13} = p_2\bar{p}_3$ ,  $u_{14} = \bar{p}_1$ ,  $u_{23} = p_1\bar{p}_3\vee\bar{p}_1p_3$ ,  $u_{24} = \bar{p}_2$ ,  $u_{34} = p_1p_3$ . Then  $Z \leq U \leq \chi(A)$ . Hence  $\chi(U) = \chi(A)$ .

6.7. Exercises.

- (1) Prove: If  $A$  is an  $n \times n$  matrix, with elements in some lattice, then  $A \vee A^2 \vee \dots \vee A^n = A \vee A^2 \vee \dots \vee A^{n+r}$ ,  $r \geq 0$ .
- (2) (a) Show that the  $n \times n$  matrices with elements in some lattice with zero element and which have zeros along the main diagonal are closed under the operation (dual to "Boolean" matrix product).

$$c_{ij} = \bigwedge_{k=1}^n (a_{ik} \vee b_{kj}).$$

- (b) Define a (binary) operation on binary relations (dual to Pierce produce) which "corresponds" to the matrix operation defined above, in the case where the elements are drawn from the two-element Boolean algebra.
- (c) Determine whether the set of connection matrices modified so as to have 0's along the main diagonal is closed under matrix product.
- (3) A permutation of a set  $S$  is a 1-1 function from  $S$  onto itself. If  $A$  is an  $n \times n$  matrix with elements in some lattice, the determinant  $|A|$  of  $A$  is defined as follows:

$$|A| = \bigvee_{\pi} a_{1\pi(1)} a_{2\pi(2)} \dots a_{n\pi(n)}$$

where  $\pi$  runs through all permutations of  $S = \{1, 2, \dots, n\}$ . Show:

- (a)  $|A| \wedge |B| \leq |A \times B|$ ;
- (b)  $A^{n-1} = M$  where  $A$  is a connection matrix and  $m_{ij}$  is the determinant of the matrix obtained from  $A$  by deleting the  $j^{\text{th}}$  row and  $i^{\text{th}}$  column. Is the assumption that  $A$  is symmetric essential?
- (4) If connection matrices are called equivalent if their characteristic matrices are equal, then it has already been shown that each equivalence class is closed under  $\vee$ ,  $\times$ . Does the same hold for  $\wedge$ ?
- (5) Let matrix  $B$  be the same as matrix  $A$  except that  $b_{14} = b_{41} = \bar{p}_1 \vee p_2 \vee \bar{p}_3$  and  $b_{24} = b_{42} = p_1 \vee \bar{p}_2 \vee \bar{p}_3$ . "Simplify"  $B$ .

7. AUTOMATA AND SEQUENTIAL CIRCUITS<sup>25,28,33,45,48,55</sup>

7.1. Definition. An automaton A is a quintuple (I, V, Q, f, g) where

- I is a nonempty finite set of objects called input states,
- V is a nonempty finite set of objects called output states,
- Q is a nonempty finite set of objects called internal states,
- f is a function (called the direct transition function) from  $I \times Q$  to Q, i.e., a function which associates with each  $i \in I$  and  $q \in Q$  an element  $f(i,q) \in Q$ , and
- g is a function from  $I \times Q$  to V (called the direct output function).

With each automaton A and internal state q of A are associated functions  $T_{A,q}^1$ ,  $T_{A,q}^2$  (respectively called transition and output functions) from finite sequences of elements of I to elements of V as follows:

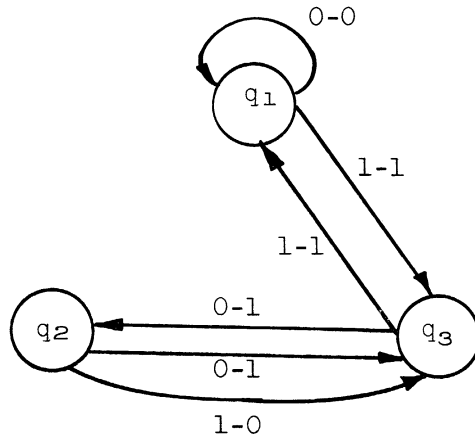
$$\begin{aligned} T_{A,q}^1(i) &= f(i,q), \\ T_{A,q}^2(i) &= g(i,q), \\ T_{A,q}^1(Si) &= f(i, T_{A,q}^1(S)), \\ T_{A,q}^2(Si) &= g(i, T_{A,q}^1(S)), \end{aligned}$$

where i is an arbitrary element of I, and S is an arbitrary (nonempty) finite sequence of elements of I. The subscript "A" will be dropped in discussions concerning a single automaton.

7.2. An example. Let  $I = \{0,1\} = V$ ,  $Q = \{q_1, q_2, q_3\}$  and f,g be given by the following table:

	f	g
0, q <sub>1</sub>	q <sub>1</sub>	0
0, q <sub>2</sub>	q <sub>3</sub>	1
0, q <sub>3</sub>	q <sub>2</sub>	1
1, q <sub>1</sub>	q <sub>3</sub>	1
1, q <sub>2</sub>	q <sub>3</sub>	0
1, q <sub>3</sub>	q <sub>1</sub>	1

Notice that the table specifies the automaton completely. The information in the table may also be presented by a direct transition-output diagram.



The label "a-b" refers to input state a and output state b. The transition and output functions associated with each of the internal states is now indicated:

Sequences of input states	$T_{q_1}^1$	$T_{q_1}^2$	$T_{q_2}^1$	$T_{q_2}^2$	$T_{q_3}^1$	$T_{q_3}^2$
0	$q_1$	0	$q_3$	1	$q_2$	1
1	$q_3$	1	$q_3$	0	$q_1$	1
0 0	$q_1$	0	$q_2$	1	$q_3$	1
0 1	$q_3$	1	$q_1$	1	$q_3$	0
1 0	$q_2$	1	$q_2$	1	$q_1$	0
1 1	$q_1$	1	$q_1$	1	$q_3$	1
0 1 1	$q_1$	1	$q_3$	1	$q_1$	1
1 0 0	$q_3$	1	$q_3$	1	$q_1$	0

7.3. It is often convenient to interpret the transition and output functions of an automaton, relative to an internal state  $q$ , in the following way. The automaton  $A$  is regarded as being in state  $q$  at "initial" time, say time  $t = 0$ . If  $T_{A,q}^1(S) = q^*$ ,  $T_{A,q}^2(S) = v$  where  $S = i_0 i_1 \dots i_n$ ,  $i_k \in I$ ,  $q, q^* \in Q$ ,  $v \in V$ , this is interpreted as meaning: if the input state of the automaton at time  $t = k$ ,  $0 \leq k \leq n$ , is  $i_k$ , then the internal state of the automaton at time  $t = n+1$  is  $q^*$  while the output state of the automaton at time  $t = n$  is  $v$ . The corresponding interpretation for the direct transition and direct output functions of an automaton is this: if at a particular time  $t$  the internal state of the automaton is  $q$  while its input state is  $i$ , then  $f(i, q)$  is the internal state of the automaton at time  $t+1$  while  $g(i, q)$  is the output state of the automaton at time  $t$ .

With this interpretation in mind, note that associated with each sequence  $i_0, i_1, \dots, i_n$  of input states and internal state  $q$  is the sequence  $q, T_{A,q}^1(i_0), \dots,$



$T_{A,q}^1(i_0i_1\dots i_{n-1})$  of internal states and  $T_{A,q}^2(i_0), T_{A,q}^2(i_0i_1), \dots, T_{A,q}^2(i_0i_1\dots i_n)$  of output states. The  $k^{\text{th}}$  member of each of the three sequences is, respectively, the input state, the internal state, and the output state of the automaton at time  $k-1$ . For the automaton in 7.2 above we indicate some sequences of input states and the associated sequences of internal and output states, assuming the automaton at time  $t = 0$  is in internal state  $q_1$ .

Input state:	1	0	1	0	1	0	1	0
Internal state:	$q_1$	$q_3$	$q_2$	$q_3$	$q_2$	$q_3$	$q_2$	$q_3$
Output state:	1	1	0	1	0	1	0	1

Input states:	0	0	1	0	0	1	0	0	1	0	0	1
Internal states:	$q_1$	$q_1$	$q_1$	$q_3$	$q_2$	$q_3$	$q_1$	$q_1$	$q_1$	$q_3$	$q_2$	$q_3$
Output states:	0	0	1	1	1	1	0	0	1	1	1	1

Input states:	1	1	0	1	0	0	0	1	1
Internal states:	$q_1$	$q_3$	$q_1$	$q_1$	$q_3$	$q_2$	$q_3$	$q_2$	$q_3$
Output states:	1	1	0	1	1	1	1	0	1

In the same way in which finite sequences of internal states and output states are associated with finite sequences of input states, one may associate infinite sequences (corresponding to times  $t = 0, 1, 2, \dots$ ) of internal and output states with infinite sequences of input states.

7.4. We now indicate how a logical net<sup>23,25</sup> may be constructed to "behave" as does a given automaton by applying the process to the example of 7.2. The first step is to "code" the input, internal, and output states. In our case, it is necessary only to "code" the set of internal states. By this is meant: associate with each internal state an ordered pair (or triple or quadruple, etc.) of zeros and ones. The ordered pairs associated with distinct internal states are required to be distinct. For example, the association

$q_1$	0 0
$q_2$	0 1
$q_3$	1 1

is permissible.

Corresponding to the table in 7.2 for  $f, g$  is the table

$x$	$a$	$b$	$a^*$	$b^*$	$y$
0,	0	0	0	0	0
0,	0	1	1	1	1
0,	1	1	0	1	1
1,	0	0	1	1	1
1,	0	1	1	1	0
1,	1	1	0	0	1

$a^*$ ,  $b^*$ ,  $y$  may be expressed as propositional formulas in  $x$ ,  $a$ ,  $b$  as follows. [Note that, since not all arguments  $x$ ,  $a$ ,  $b$  are specified in the table, more than one Boolean function (say for  $y$ ) will satisfy the conditions expressed by the table.]

$$a^* = \bar{x} \bar{a} b \vee x \bar{a} \bar{b} \vee x \bar{a} b = \bar{x} \bar{a} b \vee x \bar{a}$$

$$\boxed{a^* = \bar{a}(b \vee x)}$$

or

$$\boxed{a^* = \bar{a} b \vee \bar{a} x}$$

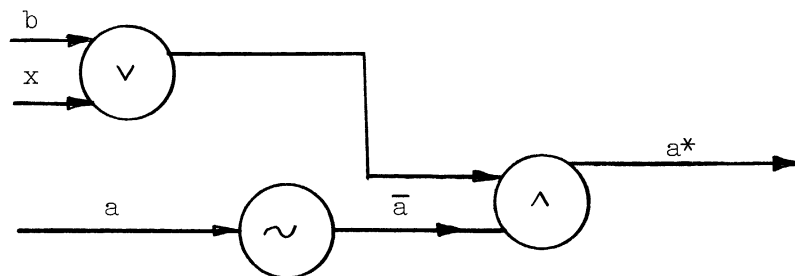
$$\begin{aligned} b^* &= \bar{x} \bar{a} b \vee \bar{x} a b \vee x \bar{a} \bar{b} \vee x \bar{a} b = \bar{a} b \vee \bar{x} a b \vee x \bar{a} \bar{b} \\ &= \bar{a} b \vee \bar{x} a b \vee x \bar{a} \\ &= \bar{a} b \vee \bar{x} b \vee x \bar{a} \\ &= \bar{x} b \vee x \bar{a} \end{aligned}$$

$$\boxed{b^* = \bar{x} b \vee x \bar{a}}$$

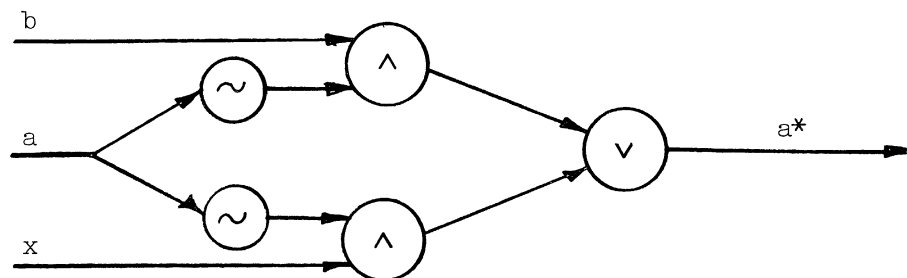
$$y = \bar{x} \bar{a} b \vee \bar{x} a b \vee x \bar{a} \bar{b} \vee x a b = \bar{x} b \vee x \bar{a} \bar{b} \vee a b$$

$$\boxed{y = b(a \vee \bar{x}) \vee \bar{a} \bar{b} x}$$

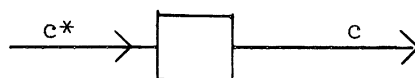
We now construct a switch(es) with inputs  $x$ ,  $a$ ,  $b$  satisfying  $a^* = a(b \vee x)$ .



or



In a similar way, switches may be constructed for  $b^*$ ,  $y$ . It is assumed that these switches act "instantaneously." In addition the net will require delay elements

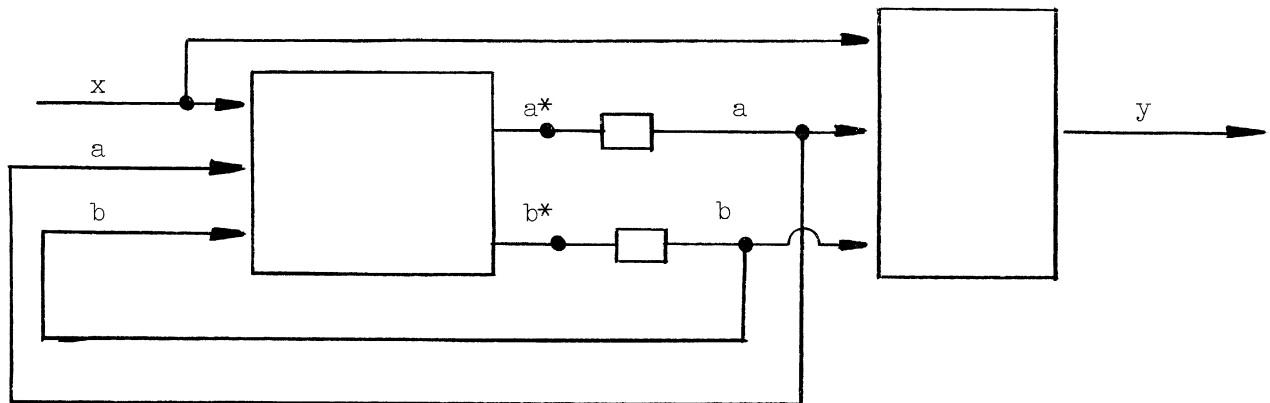


whose input and output are related as follows:

$$c^*(t) = c(t+1).$$

It is further assumed, here, that  $c(0) = 0$ .

We may now schematically indicate a net whose behavior corresponds (via the coding chosen) to the automaton. The states (i.e., 0 or 1, active or inactive) of junctions  $a$ ,  $b$  correspond to the internal states of the automaton.



The "behavior" of this net is indicated by the following equations:

$$\begin{aligned} a(t+1) &= \overline{a(t)}(b(t) \vee x(t)) \\ b(t+1) &= \overline{x(t)}b(t) \vee x(t)\overline{a(t)} \\ y(t) &= b(t)(a(t) \vee \overline{x(t)}) \vee \overline{a(t)}b(t) x(t) \\ a(0) &= 0 \\ b(0) &= 0. \end{aligned}$$

The "behavior" of this net may also be expressed by the following equations:

$$\begin{aligned} a^*(t+1) &= a^*(t)[b^*(t) \vee x(t+1)] \\ b^*(t+1) &= \overline{x(t+1)} b^*(t) \vee x(t+1) \overline{a^*(t)} \\ y(t+1) &= b^*(t)[a^*(t) \vee \overline{x(t+1)}] \vee \overline{a^*(t)} b^*(t) x(t+1) \\ a^*(0) &= x(0) \\ b^*(0) &= x(0) \\ y(0) &= x(0). \end{aligned}$$

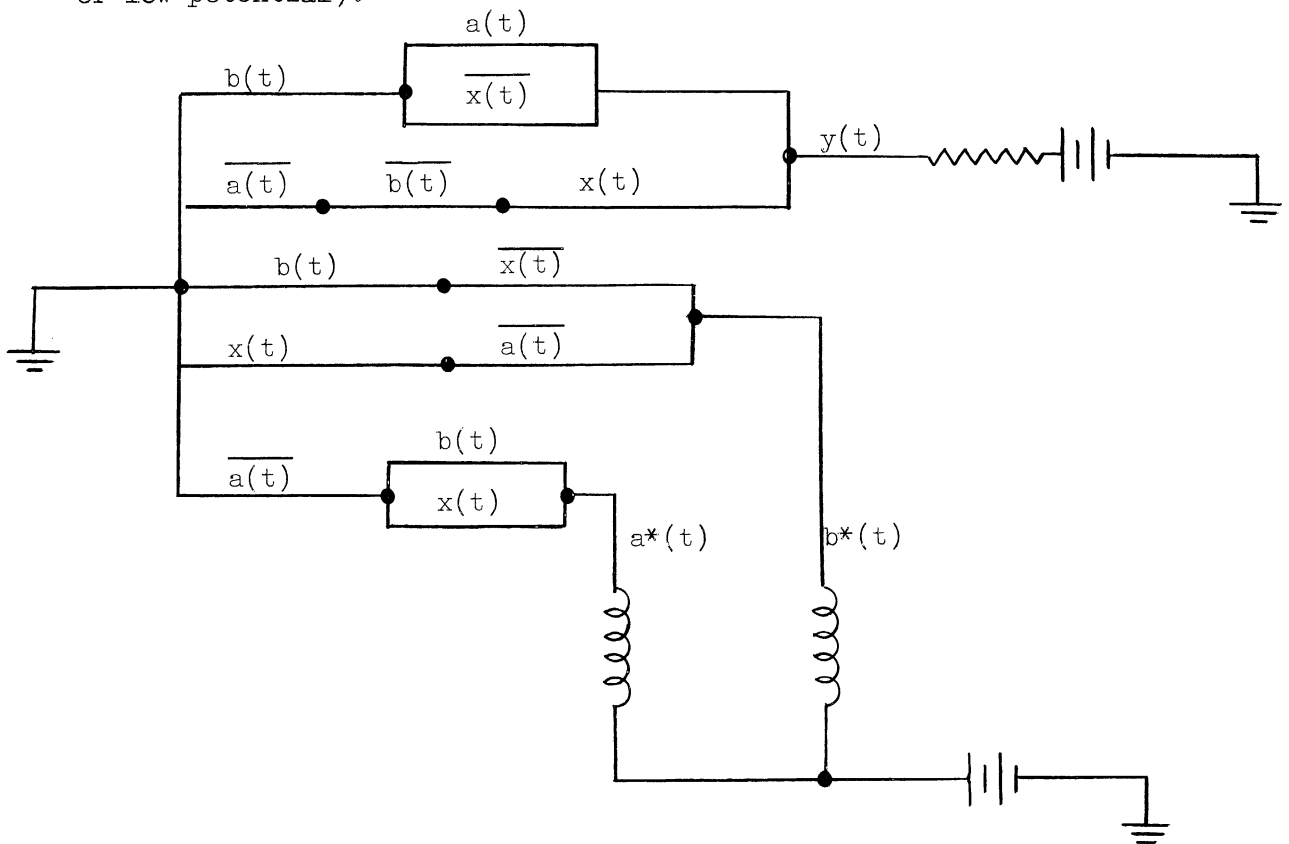
[Note that, if the internal state  $q$  and the output  $v$  of an automaton at time  $t+1$  are regarded as functionally dependent on  $q(t)$ , and the input  $i$  at  $t+1$  and further  $q(0)$  and  $v(0)$  are dependent on  $q(0)$  and  $i(0)$ , then  $(a^*, b^*)$  would correspond to the internal state of the associated automaton.]

The "input junction" labeled "x" assumes one of two (input) states. The "internal junctions" (a,b) assume one of three (internal) states, viz., 0 0, 0 1, 1 1. The "output junction" y assumes one of two (output) states. If  $x(0) = 1, x(1) = 0, x(2) = 1, x(3) = 0, x(4) = 1, \text{ etc.}$ , the corresponding sequences for a, b, y are as follows:

time	0	1	2	3	4	5	6	7
x	1	0	1	0	1	0	1	0 . . .
a	0	1	0	1	0	1	0	1 . . .
b	0	1	1	1	1	1	1	1 . . .
y	1	1	0	1	0	1	0	1 . . .

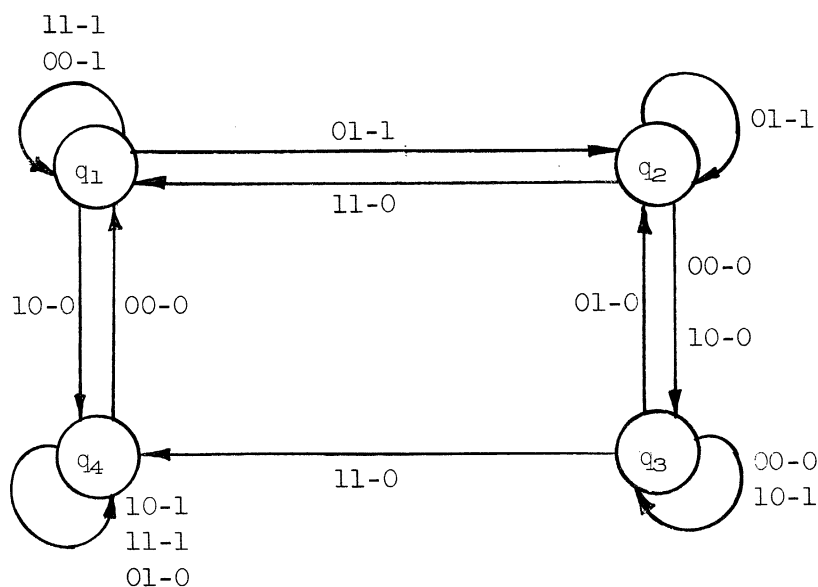
7.5 The previous section indicates how our notion of automaton is associated with synchronous sequential circuitry. We now indicate a connection with asynchronous circuitry as typified by (sequential) relay circuits. This will be done by constructing a relay circuit associated with the logical net above as follows:

- "x" will indicate the state of a relay contact whose coil is outside the circuit;
- "a\*, " "b\*" will indicate the state of coils in the circuit and
- "a, " "b" will indicate the state of the contacts which relays a\*, b\* operate;
- "y" will indicate the state of a certain point in the circuit (high potential or low potential).



There are certain difficulties with this circuit, however. These difficulties can be appreciated by reference to the direct transition-output diagram of 7.2. Let us suppose the initial input of relay contact  $x$  is 1, i.e.,  $x(0) = 1$  and the coil controlling  $x$  remains energized. Then  $(a,b)$  starts in state  $(0, 0)$ , then passes to  $(1, 1)$ , then back to  $(0, 0)$ , and back to  $(1, 1)$ , etc. Coils  $a^*$ ,  $b^*$  would oscillate back and forth between being energized and not, and the state of  $y$  would be constantly low potential.

If the coil controlling  $x$  is now de-energized, the output would depend on the states of  $a,b$  at that time which, in general, could not be predicted in advance. A relay circuit associated with the automaton below does not suffer the same difficulty. If the coding of the internal states is such that at most one relay coil changes state at any instant, then if we assume the contacts on any relay operate "simultaneously," the "behavior" of the relay circuit would correspond to the "behavior" of the automaton.



### 7.6. Exercises.

1. Construct the logical net associated with the automaton indicated above where  $q_1, q_2, q_3, q_4$  correspond, respectively, to 00, 01, 11, 10.

2. As 1 but construct a diagram of a relay circuit instead of a logical net.

3. If an automaton  $A$  has one input state, then for any internal state  $q$  there are integers  $m \geq 0, p > 0$  such that  $T_{A,q}^2(t) = T_{A,q}^2(t+p)$  for all  $t \geq m$  (cf. 7.1). The argument  $t$  represents an input sequence of length  $t$ . Further  $p$  is less than the number of internal states of the automaton.

4. Let  $f$  be any function on the non-negative integers with values in a finite set  $V$  and such that, for some  $m \geq 0, p > 0, f(t) = f(t+p)$  for all  $t \geq m$ .

Construct an automaton A with one input state, whose set of output states is V and such that  $T_{A,q}^2(t) = f(t)$  for all t, for some internal state q of the automaton A. Here, again, the argument t of the function  $T_{A,q}^2$  is to be correlated with an input sequence of length t.

### 7.7. Behavioral equivalence.

7.7.1. Two automata A,B with the same set of input and output states are behaviorally equivalent with respect to (nonempty) subsets  $Q_A, Q_B$  of the internal states of A,B [written  $(A, Q_A) \sim (B, Q_B)$ ] if and only if for each  $q_A \in Q_A$  there exists  $q_B \in Q_B$  such that (cf. 7.1)  $T_{A,q_A} = T_{B,q_B}$  and similarly for A and B reversed. Two such automata are isomorphic if and only if there exists a 1-1 function  $\phi$  from the internal states of A onto the internal states of B such that  $\phi f_A(i,q) = f_B(i, \phi(q))$  and  $g_A(i,q) = g_B(i, \phi(q))$ . An internal state  $q'$  of an automaton A is accessible from an internal state q of A if and only if  $q'$  is q or there exists a finite sequence S of input states of A such that  $T_q^1(S) = q'$ . An automaton A is minimal with respect to a subset Q of its internal states [briefly,  $(A,Q)$  is minimal] if and only if (1) for every internal state  $q'$  of A there exists an internal state  $q \in Q$  such that  $q'$  is accessible from q and (2) if  $q, q'$  are distinct internal states of A, then  $T_q^2 \neq T_{q'}^2$ . An automaton is reduced if and only if (2) holds.

7.7.2. Theorem. (a) Given any automaton A and (nonempty) subset  $Q_A$  of its internal states, there is an automaton B (whose internal states are a subset of the internal states of A) and a (nonempty) subset  $Q_B$  of its internal states such that  $(A, Q_A) \sim (B, Q_B)$  and  $(B, Q_B)$  is minimal.

(b) If  $(A, Q_A) \sim (B, Q_B)$  and  $(A, Q_A), (B, Q_B)$  are minimal, then A and B are isomorphic.

Proof. (a) Let Q be the set of internal states of A which are accessible from (elements of)  $Q_A$ . Call two elements  $q, q'$  of Q equivalent if and only if  $T_{A,q}^2 = T_{A,q'}^2$ . This binary relation is indeed an equivalence relation (cf. 5.5) and so partitions Q into equivalence classes. Call this relation R and let  $R^*(q)$  be the set of all elements  $q' \in Q$  such that  $qRq'$ , i.e., the set of all elements  $q'$  such that  $T_{A,q}^2 = T_{A,q'}^2$ . Let c be any function defined over the R-equivalence classes  $\alpha$  with the property  $c(\alpha) \in \alpha$ . The function c "chooses" from each equivalence class exactly one element from that class. We will now "construct" an automaton B to satisfy the theorem. Let the set of all internal states of B be the set of values of c, i.e., the set of all  $c(R^*(q))$  where  $q \in Q$ . Notice that  $c(R^*(q))$  is equivalent to q and, for  $q \in B$ ,  $c(R^*(q)) = q$  and  $c(R^*(q))$  and  $c(R^*(q'))$  are either equal or not equivalent. Let  $Q_B$  be the set of all  $c(R^*(q))$  where  $q \in Q_A$ . Let  $(f_A, g_A), (f_B, g_B)$  be the direct transition and output functions of A and B, respectively. Define:  $f_B(i, q') = c(R^*(f_A(i, q')))$  and  $g_B(i, q') = g_A(i, q')$  where  $q'$  is an internal state of B. The desired result will follow from the following lemma.

Lemma 1. If  $q' = c(R^*(q))$ , then  $T_{A,q}^2 = T_{B,q'}^2$  and  $c(R^*(T_{A,q}^1(S))) = T_{B,q'}^1(S)$  for all finite sequences  $S$  of input states.

Proof. Note that  $T_{A,q}^2(i) = g_A(i,q) = g_B(i,q') = T_{B,q'}^2(i)$  and

$$T_{B,q'}^1(i) = f_B(i,q') = c(R^*(f_A(i,q))) = c(R^*(T_{A,q}^1(i))) = c(R^*(T_{A,q}^1(i))).$$

Inductively assuming that the lemma holds for a finite sequence  $S$ , it is required to show that it holds for  $Si$ . Recall that  $T_{A,q}^1(S)$  is equivalent to  $c(R^*(T_{A,q}^1(S)))$ , so that  $T_{B,q'}^2(Si) = g_B(i, T_{B,q'}^1(S)) = g_B(i, c(R^*(T_{A,q}^1(S)))) = g_A(i, T_{A,q}^1(S)) = T_{A,q}^2(Si)$  and  $T_{B,q'}^1(Si) = f_B(i, T_{B,q'}^1(S)) = c(R^*(f_A(i, T_{B,q'}^1(S)))) = c(R^*(f_A(i, c(R^*(T_{A,q}^1(S))))) = c(R^*(f_A(i, T_{A,q}^1(S)))) = c(R^*(T_{A,q}^1(Si)))$ , and the lemma is proved.

Thus, given any  $q \in Q_A$ , there exists  $q' \in Q_B$ , namely,  $q' = c(R^*(q))$ , such that  $T_{A,q}^2 = T_{B,q'}^2$ . Given any  $q' \in Q_B$ , there exists  $q \in Q_A$ , namely  $q = q' = c(R^*(q'))$ , such that  $T_{A,q}^2 = T_{B,q'}^2$ , so that  $(A, Q_A) \sim (B, Q_B)$ . Furthermore,  $(B, Q_B)$  is minimal because: (1) Suppose  $q''$  is any internal state of  $B$ . There is an internal state  $q \in Q_A$  such that  $q''$  is accessible (in  $A$ ) from  $q$  since  $q'' \in Q$ . Let  $q' = c(R^*(q))$ . Then  $q' \in Q_B$ . By the lemma,  $c(R^*(T_{A,q}^1(S))) = T_{B,q'}^1(S)$ . Now  $S$  may be chosen in such a way that  $T_{A,q}^1(S) = q''$ . Since  $c(R^*(q'')) = q''$ , it follows that  $T_{B,q'}^1(S) = q''$  so that  $q''$  is accessible from an element of  $Q_B$ . (2) Let  $q, q'$  be any two internal states of  $B$ . Then  $q, q' \in Q$  and  $T_{A,q}^2 \neq T_{A,q'}^2$ . But  $q = c(R^*(q))$ ,  $q' = c(R^*(q'))$  so that, applying the lemma  $T_{B,q}^2 \neq T_{B,q'}^2$ .

We shall need the following lemma.

Lemma 2. For any automaton  $A$  and internal state  $q$  of that automaton, if  $T_{A,2}^1(S_0) = q_0$ , then, for all  $S$ ,  $T_{A,q}^1(S_0S) = T_{A,q_0}^1(S)$  and  $T_{A,q}^2(S_0S) = T_{A,q_0}^2(S)$ .

Proof. If  $S$  is  $i$ , a sequence of input states of length one, the result follows from 7.1. Now suppose the result holds for  $S'$ ,  $S$  is  $S'i$ , and  $T_{A,q}^1(S_0S') = q'$ . By inductive assumption  $q' = T_{A,q}^1(S_0S') = T_{A,q_0}^1(S')$  and  $T_{A,q}^2(S_0S') = T_{A,q_0}^2(S')$ . By 7.1,  $f(i, q') = T_{A,q}^1(S_0S'i) = T_{A,q_0}^1(S'i)$  and  $g(i, q') = T_{A,q}^2(S_0S'i) = T_{A,q_0}^2(S'i)$  and the lemma is proved.

Proof of (b). Since  $(A, Q_A) \sim (B, Q_B)$ , for  $q \in Q_A$  there exists  $q' \in Q_B$  such that  $T_{A,q}^2 = T_{B,q'}^2$ . Because of the minimal character (second defining property) of  $(B, Q_B)$ , if  $T_{A,q}^2 = T_{B,q''}^2$  also, then  $q' = q''$ . Similarly, for each  $q' \in Q_B$ , there is exactly one  $q \in Q_A$  such that  $T_{A,q}^2 = T_{B,q'}^2$ . Thus, there is a 1-1 function  $d$  from  $Q_A$  onto  $Q_B$  such that  $T_{A,q}^2 = T_{B,d(q)}^2$  for all  $q \in Q_A$ . We not extend the domain of  $d$  to all the internal states of  $A$  and its range to all the internal states of  $Q_B$ . Let  $q$  be any internal state of  $A$ . Then, because of the minimal character (first defining property) of  $(A, Q_A)$ , there exists  $S'$ ,  $q'$  such that  $T_{A,q}^1(S') = q$ . We would like to define  $d(q) = T_{B,d(q)}^1(S')$ . In order for this requirement to well-define  $d(q)$ , it must be shown that: if  $T_{A,q''}^1(S'') = q$  also, then  $T_{B,d(q'')}^1(S'') = T_{B,d(q)}^1(S')$ . Observe that  $T_{A,q'}^2(S'S) = T_{B,d(q')}^2(S'S)$

for all finite sequences of  $S$  of input states. By lemma 2  $T_{A,q}^2(S) = T_{A,q'}^2(S'S)$  and  $T_{B,q^*}^2(S) = T_{B,d(q')}^2(S'S)$  where  $q^* = T_{B,d(q')}^1(S')$ , so that  $T_{A,q}^2 = T_{B,q^*}^2$ . Similarly, if  $q^{**} = T_{B,d(q'')}^1(S'')$  then  $T_{A,q}^2 = T_{B,q^{**}}^2$ . Hence  $T_{B,q^*}^2 = T_{B,q^{**}}^2$  so that  $q^* = q^{**}$  by the minimal nature of  $B$ . Therefore,  $d(q) = T_{B,d(q')}^1(S')$  well-defines  $d(q)$ . If  $d(q) = d(q^1)$  then  $T_{A,q}^2 = T_{B,d(q)}^2 = T_{B,d(q^1)}^2 = T_{A,q^1}^2$  and so, because  $A$  is minimal,  $q = q^1$ . Thus the domain of  $d$  has been extended to the set of all internal states of  $A$  and it has been shown that  $d$  is 1-1. If  $q^*$  is any internal state of  $B$ , then for some  $S'$  and  $d(q') \in Q_B$ ,  $q^* = T_{B,d(q')}^1(S')$ . Let  $q = T_{A,q'}^1(S')$ . Then  $d(q) = T_{B,d(q')}^1(S')$  and by what has been shown above,  $T_{A,q}^2 = T_{B,d(q)}^2$ . But, because of the symmetric roles of  $A, B$ ,  $T_{A,q}^2 = T_{B,q^*}^2$  and so  $d(q) = q^*$ . Thus the range of  $d$  is the set of all internal states of  $B$ .

Corollary. If  $(A, Q_A) \sim (B, Q_B)$  and  $(B, Q_B)$  is minimal, then  $A$  has at least as many internal states as  $B$  (which indicates why the word "minimal" is used).

Proof. By (a) of the theorem, there is an automaton  $A'$  and a subset of its internal states  $Q_{A'}$  such that  $(A, Q_A) \sim (A', Q_{A'})$  and  $(A', Q_{A'})$  is minimal. Furthermore, the internal states of  $A'$  are a subset of those of  $A$ . Moreover, since  $(A', Q_{A'}) \sim (B, Q_B)$ , by (b) of the theorem, the number of internal states of  $A'$  and  $B$  are the same.

7.7.3. Exercises. 1. Show that if  $\phi$  is an isomorphism between automata  $A$  and  $B$  and if  $Q_A$  is any subset of the internal states of  $A$  and  $Q_B$  is the set of  $\phi(q)$  where  $q \in Q_A$ , then  $(A, Q_A) \sim (B, Q_B)$ .

2. Let  $K$  be the class of all automata which have the same set of two input states, two internal states, and two output states.

(a) How many elements are there in  $K$ ?

(b) How many  $(A, Q_A)$  are there where  $A \in K$ ,  $(A, Q_A)$  is minimal, and

(1)  $Q_A$  contains exactly one element,

(2)  $Q_A$  contains two elements?

(c) The relation of isomorphism between automata in  $K$  is an equivalence relation. How many equivalence classes are there?

(d) Call two automata  $A = (I, V, Q, f, g)$  and  $B = (I', V', Q', f', g')$  isomorphic in the wider sense if and only if there exists 1-1 functions  $\phi_1, \phi_2, \phi_3$  from  $I, V, Q$ , respectively, onto  $I', V', Q'$ , respectively, such that  $f'(\phi_1(i), \phi_3(f(i, q)))$  and  $g'(\phi_1(i), \phi_3(q)) = \phi_2(g(i, q))$  for all  $i \in I$  and  $q \in Q$ . The relation of isomorphism in the wider sense is an equivalence relation on the elements of  $K$ . How many equivalence classes are there? Show that, if  $A, B$  are isomorphic, then they are isomorphic in the wider sense.

7.7.4. Define a family of equivalence relations over the set of all internal states of a given automaton as follows. For each positive integer  $k$ ,  $q R_k q'$  if and only if  $T_q^2(S_k) = T_{q'}^2(S_k)$  where  $S_k$  varies through all finite sequences of input states of length  $k$  or less. Notice that, if  $q R_{k+1} q'$ , then  $q R_k q'$ . Let  $q R q'$  if and only if  $T_q^2(S) = T_{q'}^2(S)$  for all sequences  $S$  of input states (of any length). If  $q R q'$ , then  $q R_k q'$  for every  $k$ .



We now define a family  $R_k^i$  of equivalence relations over the same set of internal states of the given automaton. Let  $R_1^i = R_1$ . Assuming  $R_k^i$  has been defined, define  $R_{k+1}^i$  as follows:  $q R_{k+1}^i q'$  if and only if (1)  $q R_k^i q'$  and (2) for all input states  $i$ ,  $f(i,q) R_k^i f(i,q')$  where  $f$  is the direct transition function of the given automaton.

Lemma 1. For all  $n$ ,  $R_n = R_n^i$ .

Proof. By definition  $R_1 = R_1^i$ . Assume  $R_k = R_k^i$ ,  $k > 1$ . To prove  $R_{k+1} = R_{k+1}^i$ .  $q R_{k+1}^i q'$  if and only if  $f(i,q) R_k^i f(i,q')$  for all  $i$  if and only if  $f(i,q) R_k f(i,q')$  for all  $i$  if and only if  $T_{f(i,q)}^2(S_k) = T_{f(i,q')}^2(S_k)$  for all  $i$  and  $S_k$  if and only if  $T_q^2(iS_k) = T_{q'}^2(iS_k)$  for all  $i, S_k$  [using  $f(i,q) = T_q^1(i)$  and 7.7.2 lemma 2] if and only if  $q R_{k+1} q'$ .

Remark. By virtue of lemma 1 and the definition of  $R_k$ , it follows:  $q, q'$  are in the same  $R_{k+1}$ -equivalence class if and only if they are in the same  $R_k$ -equivalence class and  $f(i,q) = f(i,q')$  for all  $i$ , where  $f$  is the direct transition function.

Lemma 2. In any automaton, if  $q R_1 q'$  for all  $q, q'$ , then  $q R q'$  for all  $q, q'$ .

Proof. By hypothesis  $T_q^2(i) = T_{q'}^2(i)$  for all  $i$ . Assume  $T_q^2(S) = T_{q'}^2(S)$  where  $S$  is any sequence of input states, and let  $q'' = T_q^1(S)$ ,  $q''' = T_{q'}^1(S)$ . Then applying 7.7.2 lemma 2,  $T_q^2(Si) = T_{q''}^2(i)$  and  $T_{q'}^2(Si) = T_{q'''}^2(i)$ . By hypothesis  $T_q^2(Si) = T_{q'}^2(Si)$  so that  $T_{q''}^2(i) = T_{q'''}^2(i)$  and the conclusion follows.

Lemma 3. For every positive integer  $k$ ,  $R = R_k$  if and only if  $R_{k+1} = R_k$ .

Proof. If  $R = R_k$ , it is obvious that  $R_{k+1} = R_k$ . If  $R \neq R_k$ , then there exist  $q, q'$  such that  $q R_k q'$  but not  $q R q'$ . Let  $S$  be a sequence of minimum length, say  $m$ , such that  $T_q^2(S) \neq T_{q'}^2(S)$ . Then  $m > k$ . Assume  $m > k+1$ , for if  $m = k+1$  the result is immediate. Let  $S = L_{m-k-1} M_{k+1}$ , where the subscripts indicate the length of the sequence. Then using 7.7.2 lemma 2,  $T_q^2(M_{k+1}) \neq T_{q'}^2(M_{k+1})$  so that  $q'' \not R_{k+1} q'''$ . On the other hand, let  $U$  be any sequence of input states of length  $k$  or less. Then  $T_q^2(L_{m-k-1}U) = T_{q'}^2(L_{m-k-1}U)$  because  $L_{m-k-1}U$  has length less than  $m$ . Again applying 7.7.2 lemma 2,  $T_q^2(U) = T_{q'}^2(U)$  so that  $q'' R_{k+1} q'''$ .

Lemma 4. If  $R_k \neq R$ , then there are at least  $k+1$   $R_k$ -equivalence classes.

Proof. Suppose  $k = 1$  and  $R_1 \neq R$ . If there is exactly one  $R_1$ -equivalence class, i.e.,  $q R_1 q'$  for all  $q, q'$ , then by lemma 2,  $R_1 = R$ , contradicting the hypothesis. Now assume inductively  $R_m \neq R$  implies there are at least  $m+1$  equivalence classes. Suppose, now, that  $R_{m+1} \neq R$ . Then  $R_m \neq R$  and by the inductive assumption there are at least  $m+1$   $R_m$ -equivalence classes. But  $R_{m+1} \neq R_m$  because by lemma 3, if equality held, then  $R_m = R$ . Hence there are more  $R_{m+1}$ -equivalence classes than  $R_m$ -equivalence. Thus, there are at least  $m+2$   $R_{m+1}$ -equivalence classes. q.e.d.

Theorem. Given an automaton with  $n > 1$  internal states. Then for some  $k$ ,  $k \leq n-1$ ,  $R_k = R$ . In particular,  $R_{n-1} = R$ .

Proof. Applying lemma 4, there exists  $j$  such that  $R_j = R$ . Let  $k$  be the minimum of all such  $j$ 's. If  $k = 1$ , the result is immediate. Suppose  $k > 1$ . Then  $R_{k-1} \neq R$  and there are at least  $k$   $R_{k-1}$ -equivalence classes (lemma 4). If  $k > n-1$ , i.e.,  $k-1 \geq n-1$ , then there are at least  $n$   $R_{k-1}$ -equivalence classes. But there are all together only  $n$  internal states and the  $R$ -equivalence classes are obtained by partitioning each  $R_{k-1}$ -equivalence class into one or more classes. Hence  $R = R_{k-1}$ , which contradicts the choice of  $k$ . The assumption that  $k > n-1$  has led to a contradiction. Therefore,  $k \leq n-1$ .

Corollary. Given an automaton with  $n$  internal states. Suppose for some  $q, q'$  that  $T_q^2 \neq T_{q'}^2$ . Then for some sequence  $S$  of input states of length at most  $n-1$   $T_q^2(S) \neq T_{q'}^2(S)$ .

Proof. Since  $T_q^2 \neq T_{q'}^2$ ,  $q \neq q'$  and  $n > 1$  so that the theorem applies.

7.7.5. Let  $q, q'$  be distinct internal states of an automaton. To determine whether  $T_q^2 = T_{q'}^2$ , "directly" would require checking for each finite sequence  $S$  of input states whether  $T_q^2(S) = T_{q'}^2(S)$ . Since there are an infinite number of such sequences, this checking cannot be carried out. The theorem 7.7.4, however, provides a finite checking procedure, i.e., an algorithm, for determining whether  $T_q^2 = T_{q'}^2$ .

Exercise. Provide an algorithm for determining whether  $q'$  is accessible from  $q$ , for any internal states  $q, q'$  of an automaton. [Cf. 6.7 (1).]

7.7.6. If  $A = (I, V, Q, f, g)$  and  $A' = (I, V, Q', f', g')$  are two automata, their direct sum  $B = A + A'$  is the automaton  $(I, V, Q_B, f_B, g_B)$  where  $Q_B$  is the union of  $Q \times \{0\}$  and  $Q' \times \{1\}$ ,  $f_B(i, (q, 0)) = f(i, q)$ ,  $f_B(i, (q', 1)) = f'(i, q')$ ,  $g_B(i, (q, 0)) = g(i, q)$ ,  $g_B(i, (q', 1)) = g'(i, q')$ , and  $i \in I, q \in Q, q' \in Q'$ . [Recall that if  $M, N$  are sets, then  $M \times N$  is the set of ordered pairs  $m, n$  where  $m \in M, n \in N$  and  $\{0\}$  is the set whose only element is 0. If we considered only those automata  $A, A'$  such that  $Q \cap Q' = \emptyset$ , then the internal states of the direct sum automaton could be taken simply as  $Q \cup Q'$ . Notice that even if  $Q \cap Q' \neq \emptyset$ , it is nevertheless the case that  $Q \times \{0\} \cap Q' \times \{1\} = \emptyset$ .]

To obtain the direct transition-output diagram for  $B$  from the diagrams for  $A, A'$  one merely changes the label  $q$  in the diagram for  $A$  to  $(q, 0)$  and the label  $q'$  in the diagram for  $A'$  to  $(q', 1)$  and the two diagrams taken together constitute the diagram for  $B$ .

The following lemma is obvious.

Lemma. (a)  $T_{A, q}^2 = T_{B, (q, 0)}^2$  and  $T_{A', q'}^2 = T_{B, (q', 1)}^2$ . Moreover,  
 (b)  $(T_{A, q}^1(S), 0) = T_{B, (q, 0)}^1(S)$  and  $(T_{A', q'}^1(S), 1) = T_{B, (q', 1)}^1(S)$  for any sequence  $S$  of input states.

Theorem. If  $A, A'$  are automata with  $n, n'$  internal states, respectively, if  $q, q'$  are, respectively, internal states of  $A, A'$ . and if  $T_{A,q}^2 \neq T_{A',q'}^2$ , then for some finite sequence  $S$  of length  $n+n'-1$  or less,  $T_{A,q}^2(S) \neq T_{A',q'}^2(S)$ .

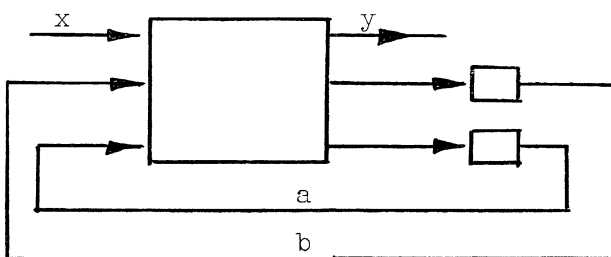
Proof. Immediate from lemma (a) immediately above and theorem 7.7.4.

Corollary. If  $A, A'$  are automata with  $n, n'$  internal states, respectively, and  $Q_A, Q_{A'}$  are subsets, respectively, of the internal states of  $A, A'$ , then whether or not  $(A, Q_A) \sim (A', Q_{A'})$  can be effectively determined. More explicitly, whether or not  $(A, Q_A) \sim (A', Q_{A'})$  can be determined by examining  $T_{A,q}^2(S), T_{A',q'}^2(S)$  for all  $q \in Q_A, q' \in Q_{A'}$  and for all sequences  $S$  of length at most  $n + n'-1$  and applying the theorem.

7.7.7. Exercises. (1) Given automata  $A, B$ . Let  $\beta = \beta(q, q')$  be a binary relation such that:  $q_1 \beta q_2$  if and only if (1) there exists  $S$  such that  $T_{A,q_1}^2(S) = q_1$  and  $T_{B,q_2}^2(S) = q_2$  or (2)  $q = q_1$  and  $q' = q_2$ . Prove:

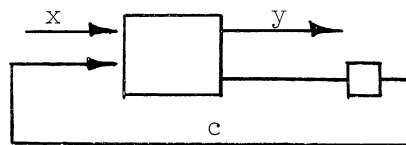
- (a) If  $T_{A,q}^2 = T_{B,q'}^2$ , then  $q_1 \beta q_2$  implies  $T_{A,q_1}^2 = T_{B,q_2}^2$ .
- (b) If  $q_1 \beta q_2$  implies  $T_{A,q_1}^2(i) = T_{B,q_2}^2(i)$  for all input states  $i$ , then  $T_{A,q}^2 = T_{B,q'}^2$ .
- (c) From (a), (b) it follows:  $T_{A,q}^2 = T_{B,q'}^2$  if and only if for all  $q_1, q_2, q_1 \beta q_2$  implies  $g_A(i, q_1) = g_B(i, q_2)$  for all  $i$ .
- (d) It is possible to decide effectively whether  $q_1 \beta q_2$  or not.

(2) Apply theorems 7.7.4 and 7.7.2 to the following problem. Given the logical net indicated below with two delay elements.



$$\begin{aligned}
 a(0) &= 0, & b(0) &= 0 \\
 a(t+1) &= \bar{x}(t)[\bar{a}(t) b(t) \vee a(t) \bar{b}(t)] \\
 b(t+1) &= \bar{x}(t) \bar{a}(t) \bar{b}(t) \\
 y(t) &= \bar{a}(t) \bar{b}(t) \vee x(t)[\bar{a}(t) b(t) \vee a(t) \bar{b}(t)]
 \end{aligned}$$

Find another (equivalent) net (of the form)



with one delay element with the property: for any sequence  $x(0), x(1), \dots, x(t)$  of (input states) of "wire"  $x$  in both nets, the determined (output) state  $y(t)$  is the same for both nets.

## 7.8 Strongly connected automata.<sup>48</sup>

7.8.1. Given any automaton  $A$ . Define  $T_{A,q}(i_1 i_2 \dots i_n)$  to be the sequence of output states

$$T_{A,q}^2(i_1), T_{A,q}^2(i_1 i_2), \dots, T_{A,q}^2(i_1 i_2 \dots i_n) .$$

As usual the subscript  $A$  will sometimes be dropped in discussions concerning only one automaton.

Lemma. For any automaton,  $T_q = T_{q'}$  if and only if  $T_q^2 = T_{q'}^2$ . Further:  $q R_k q'$  if and only if  $T_q(S) = T_{q'}(S)$  for all sequences  $S$  of length  $k$ .

Making use of 7.7.2 lemma 2, the proof follows readily.

7.8.2. An automaton is strongly connected if and only if for every ordered pair  $(q, q')$  of internal states of the automaton,  $q'$  is accessible from  $q$ . Note that, given an automaton, one can effectively determine whether it is strongly connected or not [cf. 7.7.5].

Theorem. Let  $A$  be a strongly connected automaton. Let  $B$  be any automaton. Suppose that for each internal state  $q$  of  $A$  and for each finite sequence  $S$  of input states there exists an internal state  $q'$  of  $B$  such that  $T_{A,q}(S) = T_{B,q'}(S)$ . Then for each internal state  $q$  of  $A$ , there exists an internal state  $q''$  of  $B$  such that for all finite sequences  $S$  of input states  $T_{A,q}(S) = T_{B,q''}(S)$ .

Proof. Define  $e_q(S)$  as the set of  $q'' = T_{B,q'}^1(S)$  where  $q'$  is an internal state of  $B$  satisfying  $T_{A,q}(S) = T_{B,q'}(S)$ . Let  $ce_q(S)$  be the number of elements in  $e_q(S)$ . Notice that  $ce_q(S)$  is at most equal to the number of states  $q'$  such that  $T_{A,q}(S) = T_{B,q'}(S)$ . Hence, for any sequences  $S, S'$ ,  $ce_q(SS') \leq ce_q(S)$  because if  $T_{A,q}(SS') = T_{B,q'}(SS')$ , then also  $T_{A,q}(S) = T_{B,q'}(S)$ . Indeed, the following holds:

Lemma.  $e_q(S'S)$  is the set of  $q'' = T_{B,q'}^1(S)$  where  $q' \in e_q(S')$  satisfies  $T_{A,\bar{q}}(S) = T_{B,q'}(S)$ , where  $\bar{q} = T_{A,q}^1(S')$ .

Let  $S_1$  be an arbitrary finite sequence of input states. Suppose there is some  $S$  such that  $ce_q(S_1 S) < ce_q(S_1)$ . Pick one and call it  $S_2$ . Suppose there is some  $S$  such that  $ce_q(S_1 S_2 S) < ce_q(S_1 S_2)$ . Then pick one and call it  $S_3 \dots$ . This process cannot go on indefinitely since  $ce_q(S)$  is a non-negative integer for all  $S$ . Thus, there is a  $k$  such that for all  $S$ ,  $ce_q(S_1 S_2 \dots S_k) = ce_q(S_1 S_2 \dots S_k S)$ . Let  $S^0 = S_1 S_2 \dots S_k$ . Then  $ce_q(S^0) = ce_q(S^0 S)$  for all  $S$ . Let the set of internal states of  $A$  be  $\{q_1, q_2, \dots, q_n\}$ . Since  $A$  is strongly connected, there exists a sequence, say  $S^1$ , of input states, possibly of length zero, such that  $q_1$  is accessible from  $T_{A,q}^1(S^0)$ . Then  $T_{A,q}^1(S^0 S^1) = q_1$ . In general, there are sequences  $S^j$  (some of which may have length zero),  $j = 1, 2, \dots, n$ , such that  $T_{A,q}^1(S^0 S^1 S^2 \dots S^j) = q_j$ . Moreover,  $ce_q(S^0 S^1 S^2 \dots S^j) > 0$ , for each  $j$ , since it follows from the hypothesis that for any  $S$ ,  $ce_q(S) > 0$ .

Let  $q' \in e_q(S^0S^1\dots S^j)$ . We shall show  $T_{A,q_j} = T_{B,q'}$  by supposing the contrary and deriving a contradiction. Suppose then for some  $S$ ,  $T_{A,q_j}(S) \neq T_{B,q'}(S)$ . Then  $ce_q(S^0S^1\dots S^j) < ce_q(S^0)$  since not all  $q' \in e_q(S^0S^1\dots S^j)$  satisfies (cf. the lemma)  $T_{A,q_j}(S) = T_{B,q'}(S)$  and  $ce_q(S^0) \geq ce_q(S^0S^1\dots S^j) \geq ce_q(S^0S^1\dots S^jS)$ , and this contradicts the choice of  $S^0$ .

7.8.3. Lemma. In a reduced automaton  $k < n$  implies there are at least  $k+1$   $R_k$ -equivalence classes.

Proof. It follows from the hypothesis that there are exactly  $n$   $R$ -equivalence classes. If  $R_k = R$ , then there are at least  $k+1$   $R_k$ -equivalence classes since  $k+1 \leq n$ . If  $R_k \neq R$ , then the conclusion follows from 7.7.4 lemma 4.

7.8.4. Theorem. Given a reduced automaton with  $n$  internal states. Let  $e_q(S)$  be the set of  $q'' = T_{q'}^1(S)$  where  $q'$  is such that  $T_q(S) = T_{q'}(S)$ . Then for every  $q$ , there exists  $S$  of length  $n(n-1)/2$  such that  $e_q(S)$  has exactly one element.

Proof. Let  $ce_q(S)$  be the number of elements in  $e_q(S)$ . Let  $l(S)$  be the length of  $S$ . It is to be proved that, for each  $q$ , there exists  $S$  such that  $l(S) = n(n-1)/2$  and  $ce_q(S) = 1$ . The theorem is obvious in case  $n = 1$ . Assume, therefore,  $n > 1$ .

Claim. If  $l(S) \leq k(k+1)/2$ ,  $1 \leq k \leq n-2$  and  $ce_q(S) = n-k$ , then for some  $k' > k$  there exists  $S'$ , an extension of  $S$ , such that  $l(S') \leq k'(k'+1)/2$  and  $ce_q(S') = n-k'$ . To justify the claim, note that there are at least  $k+2$   $R_{k+1}$ -equivalence classes (7.8.3). Not all  $n-k$  members of  $e_q(S)$  can be in the same  $R_{k+1}$ -equivalence class, for if this were so then the automaton would have at least  $(n-k)+(k+1)$  internal states (since each equivalence class is nonempty). It follows that there exist  $q_1, q_2$  such that  $q_1 \in e_q(S)$ ,  $q_2 \in e_q(S)$  and  $q_1 \not\sim_{k+1} q_2$ . For some  $S_{k+1}$ ,  $l(S_{k+1}) = k+1$  and  $T_{q_1}(S_{k+1}) \neq T_{q_2}(S_{k+1})$ . If  $\bar{q} = T_q^1(S)$ , then  $\bar{q} \in e_q(S)$  and either  $T_{\bar{q}}(S_{k+1}) \neq T_{q_1}(S_{k+1})$  or  $T_{\bar{q}}(S_{k+1}) \neq T_{q_2}(S_{k+1})$ . Hence, by lemma 7.8.2 (applied in the case  $A = B$ ),  $ce_q(SS_{k+1}) < ce_q(S)$ . Let  $S' = SS_{k+1}$  and choose  $k'$  such that  $ce_q(S') = n-k'$ . Then  $k' > k$ . Furthermore,  $l(S') \leq k(k+1)/2 + k+1 = (k+1)(k+2)/2 \leq k'(k'+1)/2$  and the claim is justified.

Since the automaton is reduced, there are at least 2  $R_1$ -equivalence classes. Suppose  $n = 2$  and  $T_q(i) \neq T_{q'}(i)$ . Then  $n(n-1)/2 = 1$ ,  $e_q(i) = \{T_q^1(i)\}$  so that  $ce_q(i) = 1$  and the theorem is verified in this case.

Suppose, now,  $n > 2$ . For some  $S$ ,  $q'$ ,  $l(S) = 1$ ,  $T_q(S) \neq T_{q'}(S)$ . Therefore,  $ce_q(S) \leq n-1$ . Let  $k$  be such that  $ce_q(S) = n-k$ . Then  $1 \leq n-k \leq n-1$  so that  $1 \leq k \leq n-1$ . If  $k = n-1$ , then  $ce_q(S) = 1$  and the conclusion of the theorem is satisfied. If  $1 \leq k \leq n-1$ , i.e.,  $1 \leq k \leq n-2$ , then the hypothesis of the claim is satisfied. Hence there is a  $k' > k$  and an extension  $S'$  of  $S$  such that  $l(S') \leq k'(k'+1)/2$  and  $ce_q(S') = n-k'$ . Now  $k' \neq n$  because  $T_q^1(S') \in e_q(S')$ . Therefore either  $k' = n-1$ , in which case the conclusion of the theorem is satisfied, or else  $k' \leq n-2$ , in which case the hypothesis of the claim is satisfied and the procedure is iterated, thereby obtaining a  $k'' > k' > k$  and a  $S''$  such that

$l(S'') < k''(k''+1)/2$ . Thus, for some  $n$ ,  $k^{(n)} = n-1$ ,  $l(S^{(n)}) \leq (n-1)n/2$  and  $ce_q(S^{(n)}) = 1$ . If  $S^{(n)}$  is extended in an arbitrary manner to a sequence of length  $(n-1)n/2$ , then this latter sequence satisfies the conclusion of the theorem. q.e.d.

Exercise. (1) Construct a reduced automaton  $A$  with  $I = \{0,1\} = V$  such that, for some  $q$ ,  $T_{A,q}^2(S) = 1$  for those and only those sequences  $S$  which do not consist solely of 0's. For each  $q$ , answer the question: Is  $A, \{q\}$  minimal? Is there an automaton with exactly one internal state (and the same input, output states as  $A$ ) which "behaves" in the same way? Construct an automaton  $B$  with  $\{0,1\}$  as its set of input and output states and such that, for some  $q$ ,  $T_{B,q}^2(S) = 1$  for those and only those sequences  $S$  which consist solely of 1's.

(2) Find a sequence  $S$  such that for all internal states  $q$  of  $A$  [exercise (1)]  $ce_q(S) = 1$ .

(3) The theorem fails if the automaton is not reduced. Indeed, one can find an automaton such that, for all  $q, S$ ,  $ce_q(S) > 1$ .

Construct such an automaton.

7.8.5. The following is an alternative, more perspicuous, formulation of 7.8.4.

Theorem. Given a reduced automaton with  $n$  internal states. Then

$$\bigwedge_q \bigvee_S \bigwedge_{q'} \left[ [T_{q'}^1(S) \neq T_q^1(S) \implies T_{q'}(S) \neq T_q(S)] \wedge l(S) = \frac{n(n-1)}{2} \right]$$

(i.e., for all  $q$ , there exists  $S$  such that for all  $q' \dots$ ).

Proof. It will be shown that  $ce_q(S) = 1$  if and only if  $\bigwedge_{q'} [T_{q'}(S) = T_q(S) \implies T_{q'}^1(S) = T_q^1(S)]$ .

(1) Suppose  $ce_q(S) = 1$  and  $T_{q'}(S) = T_q(S)$ . Then  $T_{q'}^1(S) \in e_q(S)$ . Since  $T^1(S) \in e_q(S)$  and  $ce_q(S) = 1$ , it follows that  $T_{q'}^1(S) = T_q^1(S)$ .

(2) Suppose  $\bigwedge_{q'} [T_{q'}^1(S) = T_q^1(S) \implies T_{q'}(S) = T_q(S)]$  and suppose  $q'' \in e_q(S)$ . Then for some  $q'$ ,  $T_{q'}^1(S) = q''$  and  $T_{q'}(S) = T_q(S)$ . It follows that  $q'' = T_{q'}^1(S) = T_q^1(S)$  so that  $e_q(S)$  has exactly one element, viz.  $T_q^1(S)$ .

The theorem is now immediate from 7.8.4.

7.8.6. The use of quantifiers permits a more perspicuous formulation of theorem 7.8.2: Let  $A$  be a strongly connected automaton and  $B$  any automaton. Then if

$$\bigwedge_q \bigwedge_S \bigvee_{q'} [T_{A,q}(S) = T_{B,q'}(S)],$$

it follows that

$$\bigwedge_q \bigvee_{q'} \bigwedge_S [T_{A,q}(S) = T_{B,q'}(S)].$$

7.8.7. Corollary. Given a reduced automaton such that

$$\bigwedge_q \bigwedge_{q'} \bigwedge_S [q \neq q' \implies T_q^1(S) \neq T_{q'}^1(S)] \text{ then } \bigwedge_q \bigvee_S \bigwedge_{q'} [(q \neq q' \implies T_{q'}(S) \neq T_q(S)) \wedge \ell(S) = n(n-1)/2].$$

Proof. Immediate from 7.8.5.

7.8.8. Corollary. If the direct sum of two automaton A and A' is reduced, then

$$\bigwedge_q \bigvee_S \bigwedge_{q'} [T_{A,q}(S) \neq T_{A',q'}(S) \wedge \ell(S) = n(n-1)/2]$$

where  $q, q'$  are internal states of A, A', respectively, and  $n$  is the number of internal states of  $A + A'$ .

Proof. [Cf. 7.7.6.]  $T_{(q,0)}^1(S) \neq T_{(q',1)}^1(S)$  and the result follows from 7.8.5.

7.8.9. Theorem. In a reduced automaton:

$$\bigvee_S \bigwedge_q \bigwedge_{q'} \left[ [T_q^1(S) \neq T_{q'}^1(S) \implies T_q(S) \neq T_{q'}(S)] \wedge \ell(S) = \frac{n(n-1)^2}{2} \right]$$

where  $n$  is the number of internal states of the automaton.

(Cf. Reference 33 for an  $S$  of shorter length than is given here. Note that in both References 33 and 48 the output state of a machine is a function only of the internal state. The notion of automaton adopted here has the output state a function of both input and internal states. One must be on guard in interpreting results established by different authors because of differences in basic concepts.)

Proof. Let  $q_1, q_2, \dots, q_n$  be the set of all internal states of the automaton. Let  $q_1^! = q_1$  and let  $S_1$  be such that

$$\bigwedge_q [T_q^1(S_1) \neq T_{q_1}^1(S_1) \implies T_q(S_1) \neq T_{q_1}(S_1)] \text{ and } \ell(S_1) = \frac{n(n-1)}{2}.$$

Suppose  $1 \leq j < n-1$  and  $q_j^!$  and  $S_j$  have been chosen such that

$$\bigwedge_q [T_q(S_j) \neq T_{q_j^!}(S_j) \implies T_q(S_j) \neq T_{q_j^!}(S_j)] \text{ and } \ell(S_j) = \frac{n(n-1)}{2}.$$

Then let  $q'_{j+1} = T_{q_{j+1}}(S_1 S_2 \dots S_j)$  and choose  $S_{j+1}$  such that

$$\bigwedge_q [T_q^1(S_{j+1}) \neq T_{q'_{j+1}}^1(S_{j+1}) \implies T_q(S_{j+1}) \neq T_{q'_{j+1}}(S_{j+1})] \text{ and } \ell(S_{j+1}) = \frac{n(n-1)}{2}.$$

Let  $S = S_1 S_2 \dots S_{n-1}$ . It is claimed that this  $S$  satisfies the requirements of the theorem. Let  $1 \leq j \leq k \leq n$  and suppose  $T_{q_j}^1(S) \neq T_{q_k}^1(S)$ . Then  $T_{q_j}^1(S S_2 \dots S_j) \neq T_{q_k}^1(S_1 S_2 \dots S_j)$ . If  $j = 1$ , then  $T_{q_k}(S_1) \neq T_{q_j}(S_1)$  so that  $T_{q_k}(S) \neq T_{q_j}(S)$ . Now suppose that  $j > 1$ . Then  $T_{q_j}^1(S_1 S_2 \dots S_j) = T_{q_j}^1(S_j)$  and  $T_{q_k}^1(S_1 S_2 \dots S_j) = T_{q_k}^1(S_j)$  where  $q = T_{q_k}(S_1 S_2 \dots S_{j-1})$  so that  $T_q(S_j) \neq T_{q'}(S_j)$ . Hence  $T_q(S_j) \neq T_{q'}(S_j)$ . Note that  $T_{q_j}(S_1 S_2 \dots S_j) = T_q(S_1 S_2 \dots S_{j-1}) T_{q_j}^1(S_j)$  and  $T_{q_k}(S_1 S_2 \dots S_j) = T_{q_k}(S_1 S_2 \dots S_{j-1}) T_{q_k}^1(S_j)$  so that  $T_{q_j}(S_1 S_2 \dots S_j) \neq T_{q_k}(S_1 S_2 \dots S_j)$ . It follows that  $T_{q_j}(S) \neq T_{q_k}(S)$ . q.e.d.

7.8.10. The direct sum of two automata has been defined (7.7.6). In an analogous way one can define the direct sum  $A = A_1 + A_2 + \dots + A_n$  of automata,  $A_1 = (I, V, Q_1, f_1, g_1)$ ,  $A_2 = (I, V, Q_2, f_2, g_2)$ ,  $\dots$ ,  $A_n = (I, V, Q_n, f_n, g_n)$ . The direct sum automaton  $A = (I, V, Q, f, g)$  has as its set of internal states  $Q = \bigcup_{j=1}^n (Q_j \times j)$  and  $f(i, (q, j)) = (f_j(i, q), j)$ ,  $g(i, (q, j)) = g_j(i, q)$  where  $i \in I$ ,  $q \in Q_j$ . It is easy to see that  $A_1 + A_2 + \dots + A_n$  is isomorphic to  $[\dots((A_1 + A_2) + A_3) + \dots + A_n]$ . Indeed,  $A$  is isomorphic to the direct sum of the automaton  $A_1, A_2, \dots, A_n$  taken in any order and any groupings. For example,

$$A_1 + A_2 + A_3 + A_4, \quad (A_3 + A_1 + A_4) + A_2, \quad (A_2 + A_3) + (A_4 + A_1)$$

are all isomorphic.

Lemma. If  $A_j$ ,  $1 \leq j \leq n$ , are strongly connected reduced automata nonisomorphic in pairs, then their direct sum is reduced.

Proof. Note that  $(A_j, Q_j)$  is minimal, where  $Q_j$  is any nonempty subset of the internal states of  $A_j$ . It follows from 7.7.2 (b) that  $A_j, A_k$ ,  $j \neq k$ , are isomorphic if and only if  $\bigvee_q \bigvee_{q'} [T_{A_j, q} = T_{A_k, q'}]$ . (Indeed if  $A_j, A_k$  are isomorphic,  $\bigwedge_q \bigvee_{q'} [T_{A_j, q} = T_{A_k, q'}]$ . Since  $A_j, A_k$  are not isomorphic,  $\bigwedge_q \bigwedge_{q'} [T_{A_j, q} \neq T_{A_k, q'}]$ . It follows that the direct sum of the  $A_j$ 's is reduced.

7.8.11. We now note some corollaries of 7.8.9.

(a) Given a reduced automaton such that

$$\bigwedge_q \bigwedge_{q'} \bigwedge_S [q \neq q' \implies T_q^1(S) \neq T_{q'}^1(S)],$$

then

$$\bigvee_S \bigwedge_q \bigwedge_{q'} \left[ (q \neq q' \implies T_{q'}(S) \neq T_q(S)) \wedge \ell(S) = \frac{n(n-1)^2}{2} \right].$$



(b) If the direct sum  $A$  of  $A_1, A_2, \dots, A_r$  is (1) reduced or (2)  $A_j$ ,  $1 \leq j \leq r$ , is strongly connected and reduced and nonisomorphic in pairs, then

$$\bigvee_S \bigwedge_q \bigwedge_{q'} [(j \neq k \implies T_{A_j, q'}(S) \neq T_{A_k, q}(S)) \wedge \ell(S) = n(n-1)^2/2]$$

where  $q', q$  are internal states of  $A_j, A_k$ , respectively, and  $n$  is the number of internal states of  $A$ .

Proof. If  $j \neq k$ , then  $T_{A, (q', j)}(S) \neq T_{A, (q, k)}(S)$  for any  $S$  and any  $q', q$  internal states of  $A_j, A_k$ , respectively. If  $A$  is reduced, the result follows from 7.8.9. If (2) holds, the result follows from the lemma and what has just been proved. q.e.d.

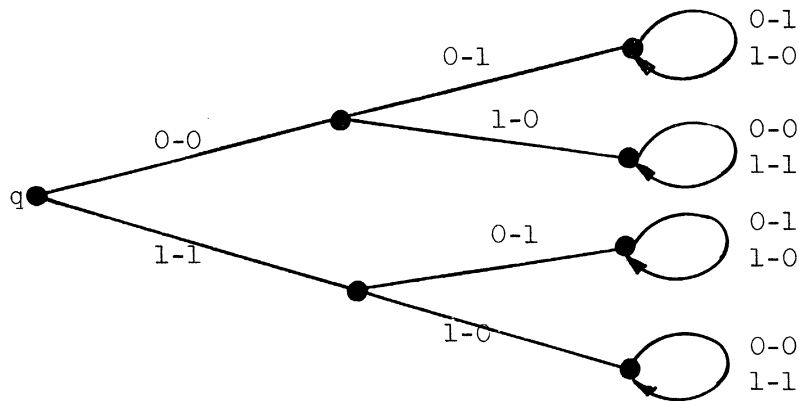
7.8.12. Theorem. Given any function  $h$  from finite sequences of elements of  $I$  of length at most  $n$ , which has values which are elements of a set  $V$ . Then there is an automaton with input states  $I$  and output states  $V$  such that for all sequences  $S$  of input states of length at most  $n$   $T_q^2(S) = h(S)$  for some internal state  $q$  of the automaton.

We illustrate a general method for constructing such an automaton, in the case  $n = 3$ ,  $I = \{0,1\} = V$  and  $h$  as indicated below.

h:	0	0
	1	1
	00	1
	01	0
	10	1
	11	0
	000	1
	001	0
	010	0
	011	1
	100	1
	101	0
	110	0
	111	1

The desired automaton is indicated by the diagram below (a modification of a transition-output diagram) where the heavy dots represent internal states. The leftmost dot plays the role of the  $q$  of the theorem.

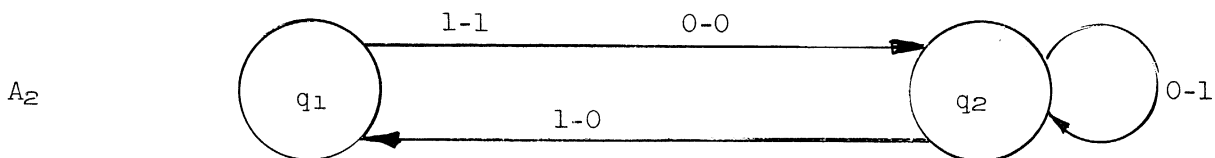
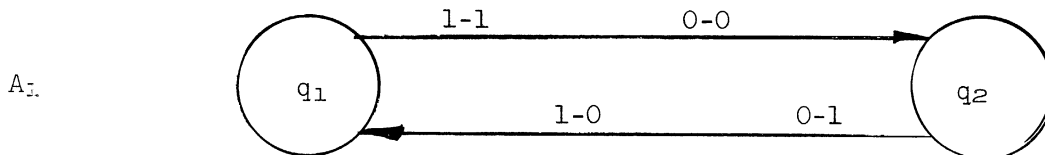
It is obvious that the indicated automaton satisfies the requirements of the theorem.



7.8.13. Exercises. (1) Knowing that the table in 7.8.12 was obtained from an automaton with two internal states, find the automaton (but do not enumerate all the two internal state automata!). Cf. 7.7.4, corollary.

(2) Call the 7 internal state automaton indicated in 7.8.12, A. Construct the minimal automaton B such that  $T_{B,q'} = T_{A,q}$  for some  $q'$  of B. Modify A by altering the transitions from the "rightmost" four internal states in such a way that the minimal "B" has two internal states.

(3) In the automata indicated below, determine whether 7.8.11 (b) (2) holds, and if so, find an S satisfying the conclusion of 7.8.11 (b).



7.9. An application of group theory. We digress to introduce some further concepts and establish some elementary mathematical results. An application is then made to automata theory.

7.9.1. The notion of "finite set" has, up to now, been presumed understood. It will be worthwhile at this point to give a precise characterization of "finite." A set  $M$  is finite if and only if every 1-1 function which takes  $M$  into itself takes  $M$  onto itself, i.e., for every function  $f$  from  $M$  into  $M$ ,

$$\left[ \bigwedge_{x,y \in M} (x \neq y \implies f(x) \neq f(y)) \implies \bigwedge_{v \in M} \bigvee_{u \in M} (v = f(u)) \right] .$$

A set of  $M$  is infinite if it is not finite, i.e., a set is infinite if and only if there exists a function  $f$  from  $M$  into  $M$  (such that)

$$\left[ \bigwedge_{x,y \in M} (x \neq y \implies f(x) \neq f(y)) \wedge \bigvee_{v \in M} \bigwedge_{u \in M} (v \neq f(u)) \right] .$$

In the case of the non-negative integers, for example,  $f$  may be chosen so that  $f(x) = x + 1$ . Alternatively,  $f$  may be chosen so that  $f(x) = 2x$ . If  $M$  is finite and  $m$  is the number of elements in  $M$ , then there are  $m^m$  functions from  $M$  into  $M$ , of which  $m!$  are 1-1 functions (and, therefore, onto). A permutation of a set  $M$  is a 1-1 function from  $M$  onto  $M$ . If  $M$  has exactly  $m$  elements, then there are  $m!$  permutations of  $M$ .

Exercise. If  $f$  is a function from a finite set onto itself, then  $f$  is a permutation. Give a counter-example in the case of an infinite set.

7.9.2. The set of all functions from a set  $A$  into a set  $B$  will be denoted by  $B^A$ . If  $f, g \in M^M$  [ $f, g$  are also called unary (singular) operations on  $M$ ], then by  $f \circ g$  is meant a function from  $M$  into  $M$  such that

$$f \circ g(x) = f(g(x)).$$

The binary operation " $\circ$ " is called "composition" and  $f \circ g$  is the result of composing  $f$  and  $g$  (in that order). In general,  $f \circ g \neq g \circ f$ . Composition is, however, associative:

$$(f \circ g) \circ h(x) = f \circ g(h(x)) = f(g(h(x))) = f(g \circ h(x)) = f \circ (g \circ h)(x).$$

Often " $fg$ " will be written instead of " $f \circ g$ ." There is a unique element, call it  $I$ , [more explicitly,  $I = I(M)$ ], satisfying  $I \in M^M$  and  $I(x) = x$  for all  $x \in M$ . For all  $f \in M^M$ ,  $fI = f = If$ .

Theorem. If  $M$  is finite,  $f, g \in M^M$ , and  $fg = I$ , then  $f$  and  $g$  are permutations and  $gf = I$ .

Proof. By hypothesis  $fg(x) = x$  for all  $x \in M$ . Suppose  $g(x) = g(y)$ . Then  $x = fg(x) = fg(y) = y$ , so that  $g$  is 1-1. Since  $M$  is finite,  $g$  is onto, i.e.,  $g$  is a permutation. For every  $x \in M$ , there exists  $y \in M$  such that  $f(y) = x$ , namely,  $y = g(x)$ . By the exercise above,  $f$  is a permutation. For every  $y \in M$ , if  $f(y) = x$ , then  $y = g(x)$ , since  $f(g(x)) = x$  and  $f$  is 1-1. It follows that  $gf(y) = g(x) = y$  for all  $y \in M$ , that is,  $gf = I$ .

Exercise. If the requirement " $M$  is infinite" is deleted from the theorem above, show the resulting statement becomes false. The following statement holds, however: If  $f, g \in M^M$  and  $fg = I = gf$ , then  $f$  and  $g$  are permutations.

7.9.3. The ordered triple  $(M, \circ, 1)$  is called a semi-group (with identity) if and only if

- (1)  $x \circ y \in M$ , for all  $x, y \in M$ ,
- (2)  $(x \circ y) \circ z = x \circ (y \circ z)$ , for all  $x, y, z \in M$ ,
- (3)  $1 \in M$  and  $1 \circ x = x = x \circ 1$ , for all  $x \in M$ .

If  $(A, \vee, \wedge)$  is a lattice with  $0, 1$  (e.g., a finite lattice), then  $(A, \wedge, 1)$  and  $(A, \vee, 0)$  are semi-groups. If  $M$  is an arbitrary set, then  $(M^M, \circ, I)$  is a semi-group. If  $P_M \subseteq M^M$  is the set of all permutations of  $M$ , then  $(P_M, \circ, I)$  is a semi-group. To justify this latter statement, it is necessary only to establish that the composition of two permutations is a permutation. Let  $f, g \in P_M$  and  $x \in M$ . Then  $f(y) = x$  and  $g(z) = y$  for some  $y$  and  $z$ . Hence  $f \circ g(z) = x$ . Suppose  $fg(x) = fg(y)$ . Then  $g(x) = g(y)$ , since  $f$  is 1-1, and  $x = y$  since  $g$  is 1-1. Thus  $fg$  is a permutation.

Lemma 1. If  $(M, \circ, 1)$  is a semi-group,  $a \in M$ , and, for all  $x$ ,  $ax = x$ , then  $a = 1$ .

Proof. Since  $ax = x$ , in particular,  $a \circ 1 = 1$ . On the other hand,  $a \circ 1 = a$ . Hence  $a = 1$ .

Lemma 2. If  $(M, \circ, 1)$  is a semi-group, then the function  $L$  taking  $a \in M$  into  $L_a$  is an isomorphism of  $(M, \circ, 1)$  and  $(\mathcal{L}, \circ, I)$  where  $\mathcal{L} \subseteq M^M$  is the set of all  $L_a$  where  $a \in M$  and  $L_a(x) = a \circ x$  for every  $x \in M$ .

Proof.  $L_a \circ L_b(x) = a \circ (b \circ x) = (a \circ b) \circ x = L_{a \circ b}(x)$ . Thus  $L_a \circ L_b = L_{a \circ b}$ .

If  $L_a = L_b$ , then  $a = L_a(1) = L_b(1) = b$ . Hence  $L$  is 1-1. By definition,  $L$  is onto. Thus  $L$  is an isomorphism.

7.9.4. A semi-group  $(M, \circ, 1)$  is a group if and only if  $\bigwedge_{x \in M} \bigvee_{y \in M} xy = 1$ . Suppose  $xy = 1 = yz$ . Then  $z = 1 \circ z = (xy)z = x(yz) = x \circ 1 = x$ . Thus, if  $xy = 1$ , then  $yx = 1$ . Suppose  $xy = 1 = xy'$ . Then  $yx = 1 = xy'$ , so that  $y = y'$ . Hence,

in a group, for each  $x$ , there exists a unique  $y$  such that  $xy = 1$ . This  $y$  satisfies  $yx = 1$  also and is denoted by " $x^{-1}$ ." In a group,  $ax = b$  has a (unique) solution for  $x$ , namely,  $x = a^{-1}b$ .

It has already been observed that  $(P_M, \circ, I)$  is a semi-group. It is also a group because if  $g$  is any permutation of  $M$ , the requirement  $fg(x) = x$  for all  $x \in M$  defines an element  $f \in P_M$ .

If  $(M, \cdot, 1)$  is a group, then  $(\mathcal{L}, \circ, I)$  is a group of permutations (cf. lemma 2, 7.9.3) and  $L_{a^{-1}} = L_a^{-1}$ .

Theorem. If  $(M, \cdot, 1)$  is a group,  $E \subseteq M$  is finite, and  $x, y, e \in E \implies x \cdot y \in E$ , then  $(E, \cdot, 1)$  is a group.

Proof. If  $a \in E$ , then  $L_a$  is a 1-1 mapping of  $E$  into  $E$ . Since  $E$  is finite,  $L_a$  is onto. Hence  $ax = a$  has a solution for  $x$  in  $E$ ; it follows  $1 \in E$ . Thus,  $(E, \cdot, 1)$  is a semi-group. Since  $L_a$  is an onto function,  $ax = 1$  has a solution in  $E$  and  $(E, \cdot, 1)$  is a group.

Corollary. If  $E$  is a finite set of permutations closed under composition (i.e.,  $f, g \in E \implies f \circ g \in E$ ), then  $(E, \circ, I)$  is a group.

Proof. If the elements of  $E$  are permutations of  $S$ , say, then  $E$  is a subset of the set  $P_S$  of all permutations of  $S$ ,  $(P_S, \circ, I)$  is a group and the theorem applies.

7.9.5. Theorem. If  $(E, \circ, I)$  is a group of permutations of  $S$ , then the relation  $\rho$  defined as:

$a \rho b$  if and only if  $a, b \in S$  and there exists  $\pi \in E$   
such that  $\pi(a) = b$  is an equivalence relation.

Proof. Since  $I \in E$ , it follows  $a \rho a$  for all  $a \in S$ . If  $a \rho b$ , then  $\pi(a) = b$  for some  $\pi \in E$  and  $a = \pi^{-1}(b)$ ,  $\pi^{-1} \in E$ , so that  $b \rho a$ . Finally, if  $a \rho b$  and  $b \rho c$ , then  $\pi(a) = b$  and  $\pi'(b) = c$  for some  $\pi, \pi' \in E$  so that  $\pi' \pi(a) = c$  and  $a \rho c$ .

7.9.6. A semi-group  $\mathcal{J}(A)$  is associated with every automaton  $A = (I, V, Q, f, g)$  in the following way. With each finite sequence  $S$  of input states, define  $L_S \in Q^Q$  as follows:  $L_S(Q) = T_q^1(S)$ . If  $\phi$  is the sequence of length 0, then  $L_\phi = I \in Q^Q$ . It is easy to verify that  $L_{SS'} = L_S \circ L_{S'}$ . Thus, if  $\mathcal{L}$  is the set of all  $L_S$ ,  $(\mathcal{L}, \circ, I)$  is a semi-group. Notice that  $\mathcal{J}(A)$  depends only on the direct transition function  $f$  (and not on the direct output function). If  $\mathcal{J}(A)$  is a group, then  $A$  is called a permutation automaton. (Cf. Ref. 33, p. 277. In Ref. 23, p. 286, def. 5, an equivalent concept "backwards deterministic" is defined.)

Lemma.  $\mathcal{J}(A)$  is a group if and only if  $L_i$  is a permutation for each input state  $i$  of  $A$ .

Proof. The composition of two permutations is a permutation (7.9.3). The result follows, then, from 7.9.4 corollary.

Theorem. Every permutation automaton A is isomorphic to a direct sum of strongly connected permutation automata.

Proof. The relation  $\rho$ , defined relative to  $\mathcal{J}(A)$ , partitions (7.9.5 theorem)  $Q$ , the internal states of A, into equivalence classes  $Q_j$ . Corresponding to each  $Q_j$  is an automaton  $A_j = (I, V, Q_j, f_j, g_j)$  where  $f_j(i, q) = f(i, q)$  and  $g_j(i, q) = g(i, q)$  for each  $i \in I, q \in Q_j$ . The isomorphism may readily be verified.

7.9.7. It has been remarked earlier that some authors use a notion of (finite) automaton in which the output at time t depends only on the internal state at time t. Let us refer to such an automaton as a Moore-machine. Let us suppose a Moore-machine given. Let the machine initially be in state q and suppose an arbitrary sequence of input states  $i_1, i_2, \dots, i_n$  given. Then the corresponding sequences of internal states and output states are given by:

$$q, T_q^1(i_1), T_q^1(i_1 i_2), \dots, T_q^1(i_1 i_2 \dots i_n)$$

and

$$g(q), g_{T_q^1}^1(i_1), g_{T_q^1}^1(i_1 i_2), \dots, g_{T_q^1}^1(i_1 i_2 \dots i_n)$$

where g is the direct output function of the automaton. Notice that the latter two sequences are of length n+1 while the former is of length n and that the initial output state is independent of the input sequences. It is natural to associate with a Moore-machine and an internal state of the machine the function  $T_q^1$ , from finite sequences S of input states to output states, defined as follows:  $T_q^1(S) = g_{T_q^1}^1(S)$ , where g is the direct output function of the machine.

Theorem 1. Suppose an automaton A and one of its internal states  $q_1$  given. Then there is a Moore-machine M with an internal state  $q_1$  such that, for all non-null S,  $T_M^1(S) = T_{A, q_1}^2(S)$ .

Proof. Let  $A = (I, V, Q, f_A, g_A)$ . Let  $M = (I, V, Q', f_M, g_M)$  where  $Q' = (I \times Q) \cup \{q_1\}$  and  $f_M(i, (i', q)) = (i, f_A(i', q))$ ,  $f_M(i, q_1) = (i, q_1)$ ,  $g_M(i, q) = g_A(i, q)$ ,  $g_M(q_1)$  may be defined arbitrarily.

We list the sequence of internal states for both automata A, M corresponding to an arbitrary sequence of input states:

	$i_1$	$i_2$	$i_3$	$i_4$	$\dots$	$i_n$
A	$q_1$	$q_2$	$q_3$	$q_4$		$q_n$ $q_{n+1}$
M	$q_1$	$(i_1, q_1)$	$(i_2, q_2)$	$(i_3, q_3)$	$(i_{n-1}, q_{n-1})$	$(i_n, q_n)$

which makes the result rather clear. Proceeding more formally, we claim that  $T_{M, q_1}^1(Si) = (i, T_{A, q_1}^1(S))$  for all input states i and sequences S thereof including

the null sequence (which has length 0). For S null, we obtain

$$T_{M,q_1}^1(i) = f_M(i, q_1) = (i, q_1) = (i, T_{A,q_1}^1(\emptyset)) .$$

Suppose

$$T_{M,q_1}^1(Si) = (i, T_{A,q_1}^1(S)) .$$

Then

$$T_{M,q_1}^1(Sii') = f_M(i', T_{M,q_1}^1(Si)) = f_M(i', (i, T_{A,q_1}^1(S))) = (i', f_A(i, T_{A,q_1}^1(S))) =$$

$$(i', T_{A,q_1}^1(Si)) ,$$

which establishes the claim.

It remains only to recall that

$$T_{A,q_1}^1(Si) = g_A(i, T_{A,q_1}^1(S))$$

and

$$T^{q_1}(Si) = g_M(T_{M,q_1}^1(Si)) = g_M(i, T_{A,q_1}^1(S)) = g_A(i, T_{A,q_1}^1(S)) . \text{ q.e.d.}$$

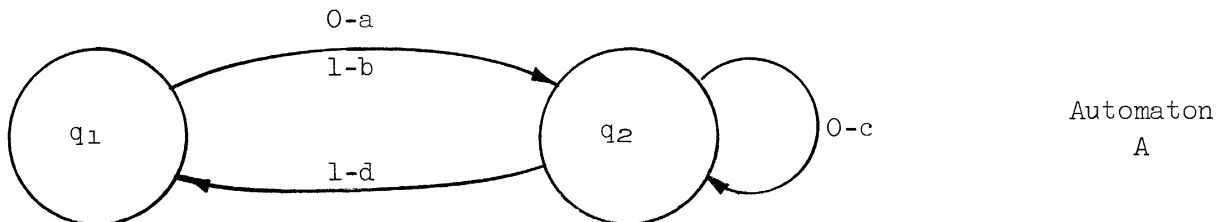
Theorem 1 states, roughly speaking, that every behavior which is realizable by a finite automaton in our sense is realizable by a Moore-machine provided, of course, "behavior," particularly of the Moore-machine, is properly understood. We now show the converse also holds. More precisely:

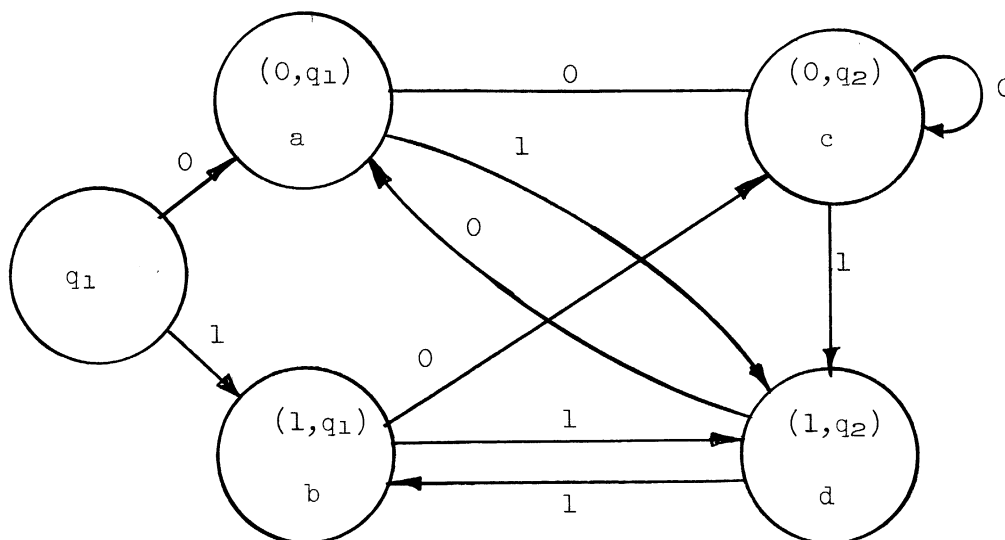
Theorem 2. Suppose that a Moore-machine  $M = (I, V, Q, f, g_M)$  with an internal state  $q_1$  given. Then if  $A = (I, V, Q, f, g_A)$ , where  $g_A(i, q) = g_M^f(i, q)$ ,  $T_M^{q_1}(S) = T_{A,q_1}^2(S)$  for all non-null S.

Proof.

$$T_{A,q_1}^2(Si) = g_A(i, T_{A,q_1}^1(S)) = g_M^f(i, T_{A,q_1}^1(S)) = g_M^{T_{M,q_1}^1}(Si) = T_M^{q_1}(Si) . \text{ q.e.d.}$$

The following two diagrams illustrate theorem 1.





Corresponding  
Moore-machine  
M

7.9.8. Exercises

1. Given an automaton A and an internal state  $q$  of that automaton. Define a binary relation  $\alpha_q$  as follows:

$$S_1 \alpha_q S_2 \text{ if and only if } T_q^1(S_1) = T_q^1(S_2).$$

Assume each internal state of A is accessible from  $q$ .

Show that:

$\alpha_q$  is an equivalence relation over the finite (including the null) sequences of input states. There is a 1-1 correspondence between internal states of the automaton and  $\alpha_q$ -equivalence classes. Treat the equivalence classes as internal states of a new automaton A' with the same input, output states as the given automaton. Define a direct transition and a direct output function for A' in such a way that A, A' are isomorphic. If the assumption concerning accessibility is dropped, what is the relationship between A, A'?

2. Define a binary relation  $\beta_q$  as follows:

$$S_1 \beta_q S_2 \text{ if and only if } T_{q_1}^2 = T_{q_2}^2 \text{ where}$$

$$q_1 = T_q^1(S_1) \text{ and } q_2 = T_q^1(S_2) \text{ .}$$

Treat the  $\beta_q$ -equivalence classes as internal states of an automaton B and define a direct transition function and a direct output function of B (as for A') in such a way that  $(A, \{q\})$  is equivalent to  $(B, \bar{q})$ , where  $\bar{q}$  is the  $\beta_q$ -equivalence class containing the null sequence. If these functions are defined in a certain "natural" way, B will be minimal.



## 8. REALIZATION OF EVENTS<sup>28,37</sup>

8.1. By an I-event is meant a set (possibly infinite) of finite sequences of elements of a finite set I. An automaton (or a logical net) with two output states 0, 1 and input states I is said to realize the I-event E (relative to one of its internal states q) if and only if (1) if  $S \in E$ , then  $T_q^2(S) = 1$  and (2) if  $T_q^2(S) = 1$ , then  $S \in E$ . The notion of "realization of an event" is another way of expressing behavior of an automaton. Certain I-events are called regular. The significance of regular events is this: every (two-output) automaton realizes a regular event and conversely, every regular event can be realized by a (two-output automaton) (cf. Refs. 7, 16).

8.2. If  $\alpha, \beta$  are I-events, then

$S \in (\alpha \vee \beta)$  if and only if  $S \in \alpha$  or  $S \in \beta$

$S \in (\alpha \cdot \beta)$  if and only if for some  $S_1, S_2, S_1 \in \alpha, S_2 \in \beta$  and  $S = S_1 S_2$

$S \in \alpha^*$  if and only if for some  $n \geq 1, S_1, S_2, \dots, S_n, S_i \in \alpha$  and  $S = S_1 S_2 \dots S_n$ .

The class of regular I-events is the smallest class containing the empty set and the unit sets of length one sequences of elements of I, which is closed under  $\vee, \cdot, *$ . Regular I-events may be denoted by regular I-expressions; these are well-formed formulas constructed out of symbols denoting the empty set, the unit sets of length one sequences of elements of I and  $\vee, \cdot, *$ . (Different regular expressions may denote the same event.) For example, letting  $I = \{0,1\}$  and letting "0," "1" denote, respectively, the unit sets  $\{0\}, \{1\}$ ,

then  $(0 \vee 1)^*$  is the set of all  $\{0,1\}$  - sequences;

$(0 \cdot 1)^*$  is the set of all  $\{0,1\}$  - sequences, beginning with 0, terminating with 1 and alternating between 0,1;

$(0.1 \vee 0.0.0)$  is the set consisting of the two sequences, 01, 000;

$1.(0 \vee 1)^*$  is the set of all sequences of length two or more which begin with a 1;

$(0^*.1.0^* \vee 1.0^* \vee 0^*.1 \vee 1)$  is the set of all sequences containing exactly one 1.

If  $\alpha$  is an I-event, then let  $[\alpha]$  be the  $I \times \{0,1\}$  event defined as follows:

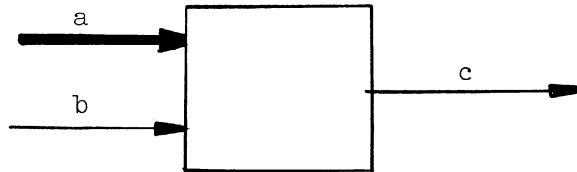
$i_1 i_2 \dots i_n$   
 $e[\alpha]$  if and only if for some k,  $p_k = 1$  and  $i_k i_{k+1} \dots i_n \in \alpha$  ;  
 $p_1 p_2 \dots p_n$

here we have written " $p_j^i$ " for the ordered pair  $(i_j, p_j)$  where  $i_j \in I$  and  $p_j \in \{0,1\}$ .

Theorem. There is a logical net (and hence an automaton) realizing  $[\alpha]$  for every regular I-event  $\alpha$ .

Proof. The proof is by induction on the number of operation symbols occurring in a regular expression A.

(a) If A is  $i$ , denoting  $\alpha = \{i\}$ , then a net realizing  $[\alpha]$  is



The set I is the set of all input states which "cable" a may assume, while the "wires" b, c are capable of assuming the states 0, 1. The switch indicated is such that for all t,

$$c(t) = 1 \text{ if and only if } a(t) = i \wedge b(t) = 1.$$

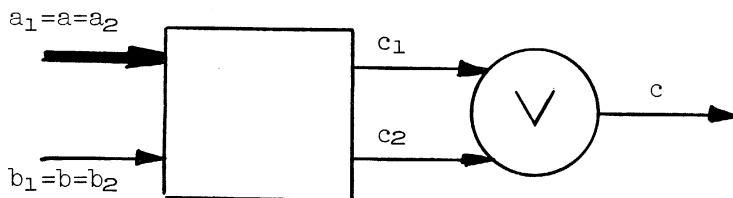
Hence this net realizes  $[\alpha]$ .

Let the expression A,  $A_1, A_2$  denote the events  $\alpha, \alpha_1, \alpha_2$ . Suppose the following nets realize  $[\alpha_1], [\alpha_2]$ , respectively.



$$\text{and } \bigwedge_t \left[ c_j(t) = 1 \iff \bigvee_{t'} [t' \leq t \quad b_j(t') = 1 \wedge a_j(t') a_j(t'+1) \dots b_j(t) \in \alpha_j] \right].$$

(b) Let  $A = (A_1 \vee A_2)$ ; then the net



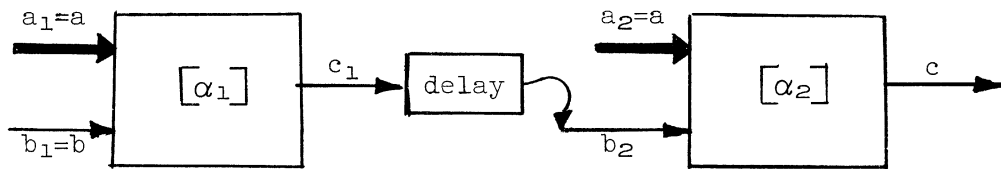
realizes  $[\alpha_1 \alpha_2]$  because  $c(t) = 1 \iff c_1(t) = 1 \vee c_2(t) = 1$  so that

$$\bigwedge_t \left[ c(t) = 1 \iff \bigvee_{t',j} [t' \leq t \wedge b_j(t') = 1 \wedge a_j(t')a_j(t'+1) \dots a_j(t) \in \alpha_j] \right]$$

and since  $\bigwedge_t [a_1(t) = a(t) = a_2(t) \wedge b_1(t) = b(t) = b_2(t)]$ ,

it follows  $\bigwedge_t [c(t) = 1 \iff \bigvee_{t'} [t' \leq t \wedge b(t') = 1 \wedge a(t')a(t'+1) \dots a(t) \in \alpha]]$ .

(c) Let  $A = (A_1 \circ A_2)$ . Then the net



realizes  $[\alpha] = [\alpha_1 \circ \alpha_2]$ . We proceed to justify this assertion. Suppose

$$\bigvee_{t'} [t' \leq t \wedge b(t') = 1 \wedge a(t')a(t'+1) \dots a(t) \in \alpha] ;$$

then there exists  $t''$ ,  $t' \leq t'' < t$ , such that

$$a(t') \dots a(t'') \dots a(t) = a(t')a(t'+1) \dots a(t) ,$$

$$a(t') \dots a(t'') \in \alpha_1 \text{ and } a(t''+1) \dots a(t) \in \alpha_2 .$$

Hence

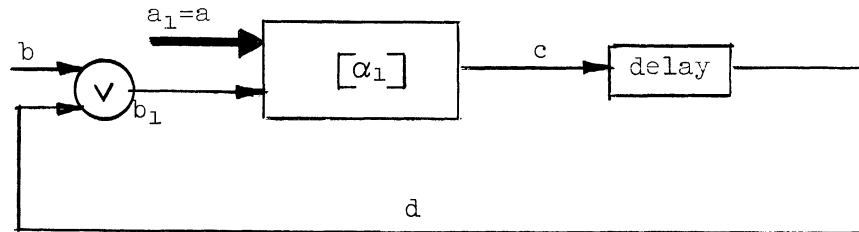
$$a_1(t') \dots a_1(t'') \in \alpha_1 \text{ and } b_1(t') = 1 \text{ so that}$$

$$c_1(t'') = 1. \text{ It follows that}$$

$$b_2(t''+1) \text{ and } a_2(t''+1) \dots a_2(t) \in \alpha_2 \text{ so that } c(t) = 1.$$

Suppose now  $c(t) = 1$ . Then for some  $t'$ ,  $t' \leq t$ ,  $b_2(t') = 1$  and  $a_2(t') \dots a_2(t) \in \alpha_2$ . Since  $b_2(t') = 1$ ,  $t' \neq 0$ . It follows  $c_1(t'-1) = 1$ . Hence there exists  $t'' \leq t' - 1$  such that  $b_1(t'') = 1$  and  $a_1(t'') \dots a_1(t'-1) \in \alpha_1$ . It follows that  $b(t'') = 1$  and  $a(t'') \dots a(t'-1)a(t') \dots a(t) \in \alpha = \alpha_1 \cdot \alpha_2$  and the assertion has been justified.

(d) Let  $A = A_1^*$ . Then the net

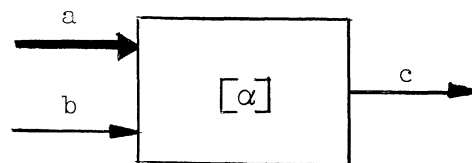


realizes  $[\alpha] = [\alpha_1^*]$ . We proceed to justify this assertion. Suppose for some  $t' \leq t$ ,  $b(t') = 1$  and  $a(t') \dots a(t) \in \alpha$ . Then, for some  $n$ ,  $t_1, t_2, \dots, t_n$ ,  $n \geq 1$ ,  $t_1 = t'$ ,  $t_1 \leq t_2 \leq \dots \leq t_n \leq t$ ,  $a(t_1)S_1a(t_2)S_2a(t_n)S_na(t) = a(t') \dots a(t)$  and  $a(t_1)S_1 \in \alpha$ ,  $a(t_2)S_2 \in \alpha$ ,  $a(t_n)S_na(t) \in \alpha$ . Since  $b(t_1) = 1$ , we have  $b_1(t_1) = 1$  and  $a(t_1)S_1 \in \alpha$ . Hence  $d(t_2) = 1$  and  $b(t_2) = 1$ . It follows  $d(t_3) = 1$ . Similarly,  $d(t_n) = 1$  and  $b_1(t_n) = 1$ . Since also  $a(t_n)S_na(t) \in \alpha$ , it follows  $c(t) = 1$ .

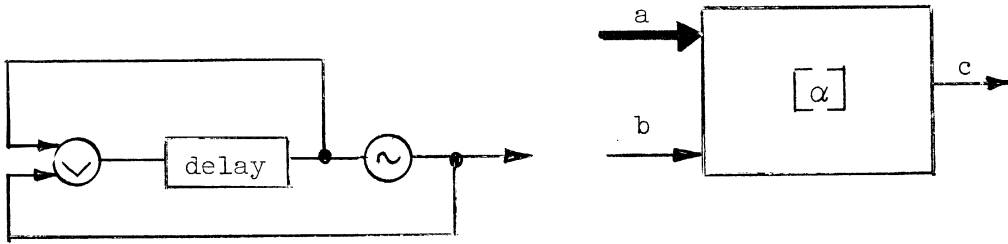
Suppose  $c(t) = 1$ . Then for some  $t' \leq t$ ,  $a_1(t') \dots a_1(t) \in \alpha_1$  and  $b_1(t') = 1$ . Hence either (1)  $b(t') = 1$  and  $a(t') \dots a(t) \in \alpha_1^*$  or (2)  $b(t') \neq 1$  and  $d(t') = 1$  so that  $t' \neq 0$ . Assuming the latter alternative holds, it follows  $c(t'-1) = 1$  so that for some  $t_1 < t'$ ,  $a_1(t_1) \dots a_1(t'-1) \in \alpha_1$  and  $b_1(t_1) = 1$ . Hence either (1)  $b(t_1) = 1$  or (2)  $b(t_1) \neq 1$  and  $d(t_1) = 1$  so that  $t_1 \neq 0$ . In case (2), it follows  $c(t_1-1) = 1$  so that for some  $t_2 < t_1$ ,  $a_1(t_2) \dots a_1(t_1-1) \in \alpha_1$  and  $b_1(t_2) = 1$ . Since there is no infinite sequence of nonnegative integers  $t \geq t' \geq t_1 > t_2 > \dots$ , it follows for some  $n$ , the first alternative will hold. For such an  $n$ , then, we have

$b(t_n) = 1$  and  $a_1(t_n) \dots a_1(t_{n-1}-1) \in \alpha_1$  so that  $a(t_n) \dots a(t) \in \alpha$ . q.e.d.

If the net



realizes  $[\alpha]$ , then the net



realizes  $\alpha$  because  $b(0) = 1$  and  $b(t+1) = 0$  so that

- (1) if  $a(0) a(1) \dots a(t) \in \alpha$ , then, since  $b(0) = 1$ ,  $c(t) = 1$ ;
- (2) if  $c(t) = 1$ , then for some  $t' \leq t$ ,  $a(t') \dots a(t) \in \alpha$  and  $b(t') = 1$ .

Hence  $t' = 0$  and  $a(0) a(1) \dots a(t) \in \alpha$ .

Corollary. There is an effective procedure for constructing a net which realizes a regular event given by a regular expression.

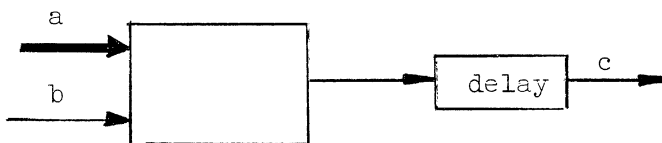
It is to be emphasized that the theorem gives a purely rote procedure for constructing a net (and hence automaton) which manifests any desired automaton behavior.

We leave the converse theorem to the bibliographical references.

8.3. We digress in this section to indicate a variant of the point of view adopted.

8.3.1. A brief demonstration is given of the theorem of the previous section in terms of Moore-machines. To motivate the construction, constructions are given first in terms of nets.

(a)



The switch is such that

$$c(t+1) = 1 \text{ if and only if } a(t) = 1 \wedge b(t) = 1.$$

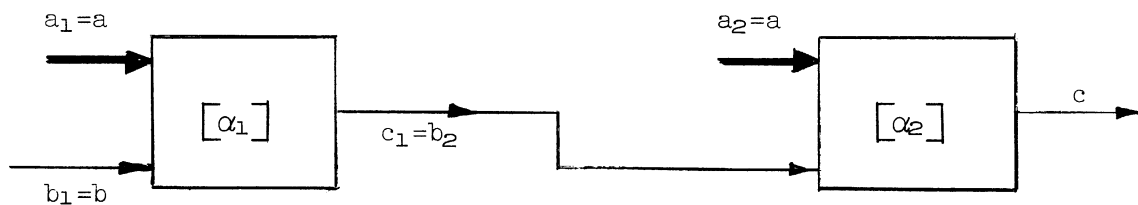
Furthermore,  $c(0) = 0$ .

Suppose the following nets given:



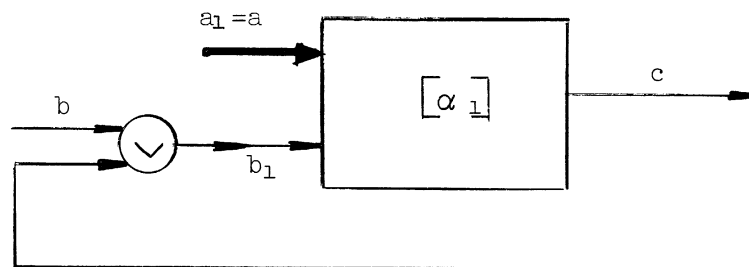
and  $c_i(t+1) = 1$  if and only if  $a_i(t_0)a_i(t_0+1) \dots a_i(t) \in \alpha_i \wedge b_i(t_0) = 1$ , and  $c_i(0) = 0$ .

(b) Then the net below with output  $c$  realizes  $[\alpha_1 \cdot \alpha_2]$ .



If  $b(t_0) = 1 \wedge a(t_0) a(t_0+1) \dots a(t) \in \alpha = \alpha_1 \cdot \alpha_2$ , then  $b_1(t_0) = 1$  and for some  $t_1$ ,  $t_0 < t_1 \leq t$ ,  $a(t_0) A_0 a(t_1) A_1 a(t) = a(t_0) a(t_0+1) \dots a(t) \wedge a_1(t_0) A_0 \in \alpha_1 \wedge a_2(t_1) A_1 a_2(t) \in \alpha_2$ , so that  $c_1(t_1) = 1 = b_2(t_1)$ . Hence  $c(t+1) = 1$ . If  $c(t+1) = 1$ , then for some  $t_1$ ,  $t_1 \leq t$ ,  $a_2(t_1) \dots a_2(t) \in \alpha_2 \wedge b_2(t_1) = 1$ . Since  $c_1(0) = 0 = b_2(0)$ , it follows  $t_1 > 0$ . Thus, for some  $t_0 < t_1$ ,  $a_1(t_0) \dots a_1(t_1-1) \in \alpha_1 \wedge b_1(t_0) = 0$ . It follows  $a(t_0) \dots a(t_1-1) a(t_1) \dots a(t) \in \alpha \wedge b(t_0) = 1$ . Note, too, that  $c(0) = 0$ .

(c) The following net realizes  $[\alpha] = [\alpha_1^*]$ , i.e., has the property  $c(t+1) = 1$  if and only if  $b(t_0) = 1 \wedge a(t_0) \dots a(t) \in \alpha = \alpha_1^*$  for some  $t_0 \leq t$ .



Suppose  $b(t_0) = 1 \wedge a(t_0) \dots a(t) \in \alpha$  for some  $t_0 \leq t$ . Then  $a(t_0) \dots a(t) = a(t_0) A_0 a(t_1) A_1 \dots a(t_n) A_n a(t)$  for some  $n, t_j$  and  $a(t_0) A_0, a(t_1) A_1, \dots, a(t_n) A_n a(t) \in \alpha_1$ . Since  $a_1(t) = a(t)$  and since  $b_1(t_0) = 1$ , it follows  $c(t_1) = 1$  so that  $b_1(t_1) = 1$  also. Similarly,  $c(t_2) = 1, b_1(t_2) = 1, \dots$ , and  $c(t_n) = 1, b_1(t_n) = 1$ . Since  $a_1(t_n) A_n a(t) \in \alpha_1$ , it follows  $c(t+1) = 1$ .

Suppose  $c(t+1) = 1$ . Then for some  $t_1 \leq t, a(t_1) \dots a(t) \in \alpha_1$  and  $b_1(t_1) = 1$ . Then either (1)  $b(t_1) = 1$  or (2)  $c(t_1) = 1$ . If (1) holds, the argument is completed. If (1) fails to hold but (2) holds, then  $t_1 > 0$  and for some  $t_2 < t_1, a(t_2) \dots a(t_1-1) \in \alpha_1$  and  $b_1(t_2) = 1$ . Again one of two alternatives holds and the argument proceeds as before.

We now give the "corresponding" constructions for Moore-machines.

(a) Suppose  $\alpha = \{i\}$ . Let  $Q_{[\alpha]} = \{0,1\}$ . Define

$$f_{[\alpha]}(i', a), q = 1 \text{ if and only if } i' = i \text{ and } a = 1,$$

$$g_{[\alpha]}(q) = q.$$

We take  $q = 0$  as initial state.

(b) Suppose  $\alpha = \beta \cdot \gamma$ . Let  $Q_{[\alpha]} = Q_{[\beta]} \times Q_{[\gamma]}$ . Define

$$f_{[\alpha]}(i, x, q, q') = \left( f_{[\beta]}(i, x, q), f_{[\gamma]}(i, g_{[\beta]}(q), q') \right)$$

$$g_{[\alpha]}(q, q') = g_{[\gamma]}(q'),$$

where  $i \in I, x \in \{0,1\}, q \in Q_{[\beta]}, q' \in Q_{[\gamma]}$ .

If  $q_{[\beta]}$  is the initial state for the automaton realizing  $[\beta]$  and similarly for  $q_{[\gamma]}$ , then we take as initial state  $q_{[\alpha]} = (q_{[\beta]}, q_{[\gamma]})$ .

(c) Suppose  $\alpha = \beta^*$ . Let  $Q_{[\alpha]} = Q_{[\beta]}$ . Define

$$f_{[\alpha]}(i, x, q) = f_{[\beta]}(i, g_{[\beta]}(q) \vee x, q),$$

$$g_{[\alpha]}(q) = g_{[\beta]}(q).$$

The initial state of the machine "for  $[\alpha]$ " is taken as the initial state of the machine for  $[\beta]$ .

(d) Suppose  $\alpha = \beta \vee \gamma$ . Let  $Q_{[\alpha]} = Q_{[\beta]} \times Q_{[\gamma]}$ .

$$f_{[\alpha]}(i, x, q, a') = \left( f_{[\beta]}(i, x, q), f_{[\gamma]}(i, x, q') \right),$$

$$g_{[\alpha]}(q, q') = g_{[\beta]}(q) \vee g_{[\gamma]}(q').$$

This initial state is chosen as in (b).

(e) Given a machine "for  $[\alpha]$ ," we now construct one for  $\alpha$ . Let  $Q_\alpha = Q_{[\alpha]} \times \{0,1\}$ . Define

$$f_\alpha(i, q, x) = \left( f_{[\alpha]}(i, x, q), 0 \right),$$

$$g_\alpha(q, 1) = 0, \quad g_\alpha(q, 0) = g_{[\alpha]}(q).$$

The initial state  $q_\alpha$  is chosen to be  $(q_{[\alpha]}, 1)$ .

8.3.2. Let a  $(\Lambda, I)$ -event be any set of I-sequences, possibly including the null sequence  $\Lambda$ . The class of  $(\Lambda, I)$ -regular events is the smallest class of  $(\Lambda, I)$ -events containing  $\emptyset, \{\Lambda\}, \{i\}$  for each  $i \in I$  and closed under  $\vee, \cdot, \dagger$  where  $S \in \alpha^\dagger$  if and only if  $S = \Lambda$  or, for some  $n, S \in \alpha^n$ . Then, since  $\alpha \cdot \alpha^\dagger = \alpha^*$  ( $= \alpha^\dagger \cdot \alpha$ ), it follows if an I-event is regular, it is  $(\Lambda, I)$ -regular.

If  $\alpha, \beta$  are sets,  $\alpha - \beta = \alpha \cap \bar{\beta}$  is a set of elements in  $\alpha$  but not in  $\beta$ . If  $\alpha$  is a  $(\Lambda, I)$ -event, let  $\alpha^0 = \alpha - \{\Lambda\}$ . We wish to show the class of  $(\Lambda, I)$ -regular events consists of exactly those  $(\Lambda, I)$ -events  $\alpha^0$  is I-regular. If  $\alpha$  is I-regular,  $\alpha \vee \{\Lambda\}$  is  $(\Lambda, I)$ -regular. We now show if  $\alpha$  is  $(\Lambda, I)$ -regular, then  $\alpha^0$  is I-regular. This follows from the following equalities:

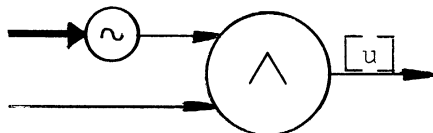
$$(\alpha \vee \beta)^0 = \alpha^0 \vee \beta^0, \quad (\alpha \cdot \beta)^0 = \alpha^0 \cdot \beta^0 \vee (\alpha - \alpha^0) \cdot \beta^0 \vee \alpha^0 \cdot (\beta - \beta^0) \quad [\text{note } \alpha \cdot \emptyset = \emptyset = \emptyset \cdot \alpha],$$

$$(\alpha^\dagger)^0 = (\alpha^0)^*.$$

The  $(\Lambda, I)$  event realized by a (two-output) Moore-machine is the I-event realized by the machine augmented by  $\Lambda$  if and only if the machine has output 1 at initial time. It follows from the underlined statement above and 8.3.1 that every  $(\Lambda, I)$  event is realizable by a Moore-machine. It follows, too, that every  $(\Lambda, I)$ -event realized by a Moore-machine is  $(\Lambda, I)$ -regular

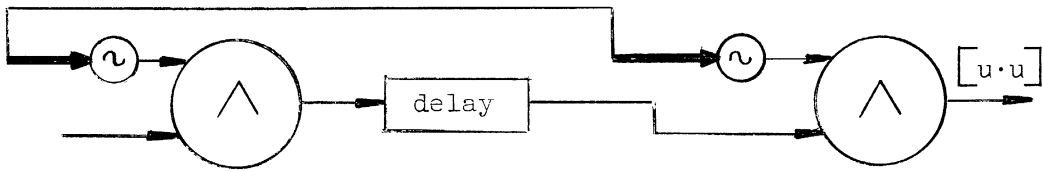
8.4. An example. We illustrate the theorem of 8.2. Let the given  $\{0,1\}$ -regular set be  $(u \cdot u \vee v)^*$  where "u" denotes  $\{0\}$  and "v" denotes  $\{1\}$ . The diagrams below indicate the application of the algorithm.

1.





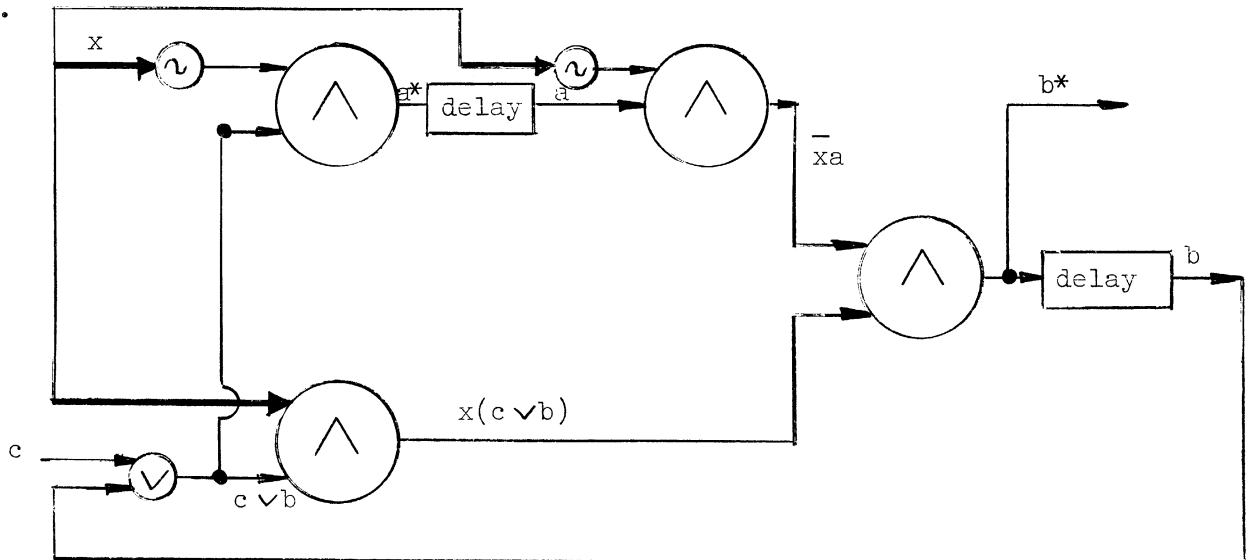
2.



3.



4.



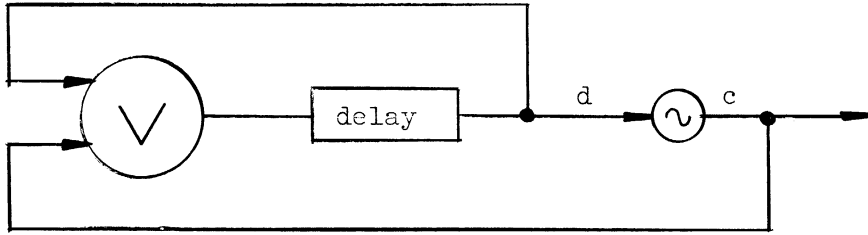
The behavior of the output  $b^*$  of this net may be described as follows:

$$(1) \quad a^* = \bar{x}(c \vee b), \quad b^* = \bar{x}a \vee x(c \vee b) .$$

[Here, the argument "t" is to be understood throughout.]

$$(2) \quad b(t+1) = b^*(t) \quad , \quad b(0) = 0, \\ a(t+1) = a^*(t), \quad a(0) = 0.$$

Assume now wire c of the net below connected to wire c of the net above.



Call the resultant net N. N may be regarded as an automaton with 8 internal states (viz., the states associated with wires a, b, d), 2 outputs states (viz., the states of wire b\*), 2 input states (viz., the states of "cable" x), and direct transition and output functions given by the table below.

To facilitate construction of the table, we set down the appropriate equations.

1. Equations for direct transition functions:

$$(i) \quad a(t+1) = \bar{x}(t) (\bar{d}(t) \vee b(t)) ,$$

$$(ii) \quad b(t+1) = \bar{x}(t) a(t) \vee x(t) (\bar{d}(t) \vee b(t)) ,$$

$$(iii) \quad d(t+1) = 1 .$$

2. Equation for direct output function:

$$b^*(t) = \bar{x}(t) a(t) \vee x(t) (\bar{d}(t) \vee b(t)) .$$

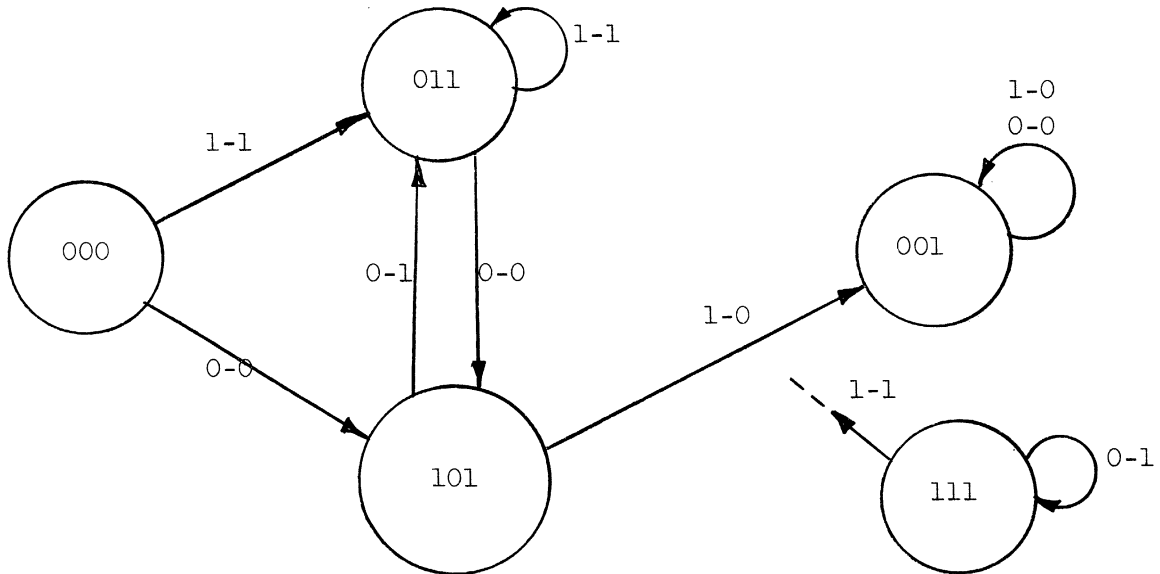
3. Equations for initial internal state:

$$a(0) = b(0) = d(0) = 0 .$$

It is clear that the states (a, b, d) = (1, 1, 0), (1, 0, 0), (0, 1, 0) are not accessible from (0, 0, 0) and so these states are omitted from the table below.

I x	Q			Q			V
	a	b	d	a	b	d	b*
0	0	0	0	1	0	1	0
0	0	0	1	0	0	1	0
0	0	1	1	1	0	1	0
0	1	0	1	0	1	1	1
0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1
1	0	0	1	0	0	1	0
1	0	1	1	0	1	1	1
1	1	0	1	0	0	1	0
1	1	1	1	0	1	1	1

Below is a direct transition-output diagram corresponding to the table above.

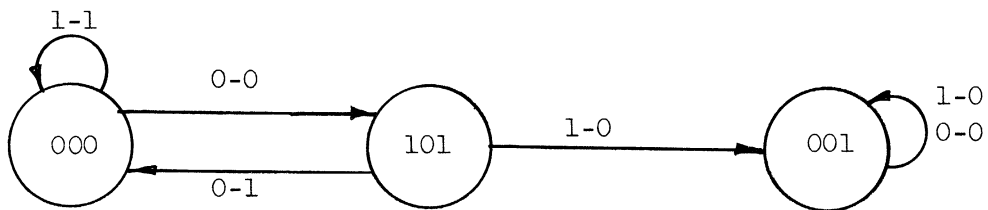


Notice that internal state 111 is not accessible from state 000. We therefore delete 111 from the set of internal states. We now partition the four internal states into equivalence classes.

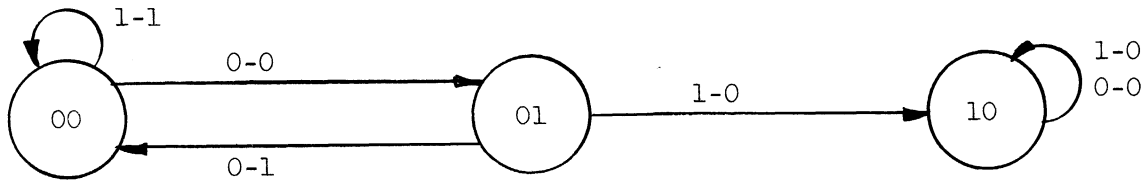
$$R_1: \{000, 011\}, \{101\}, \{001\}$$

$$R_2: \{000, 011\}, \{101\}, \{001\}$$

Thus the minimal automaton equivalent (relative to {000}) to the given automaton is given by the diagram below.

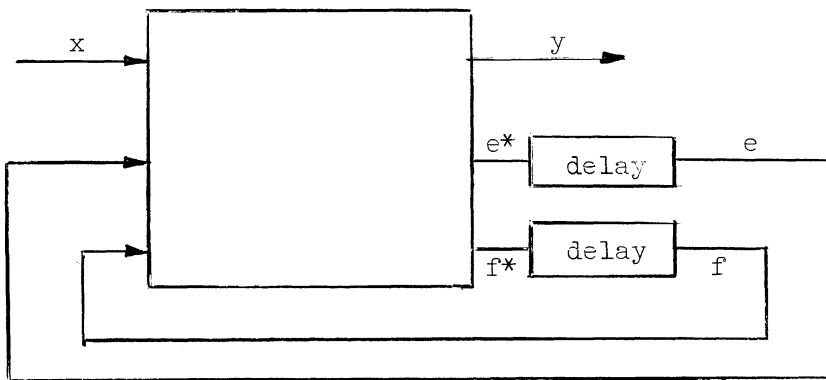


One may check at this stage that this automaton realizes the event in question. We now rename the internal states and then construct a net with an output which behaves like  $b^*$  above.



x	e	f	e f	y
0	0	0	0 1	0
0	0	1	0 0	1
0	1	0	1 0	0
1	0	0	0 0	1
1	0	1	1 0	0
1	1	0	1 0	0

$f^* = \bar{x} \bar{e} \bar{f}$ ,  $e^* = \bar{x} e \bar{f} \vee x \bar{e} f \vee x e \bar{f} = e \bar{f} \vee x \bar{e} f$ . To simplify  $e^*$ , we may require that  $e^*$  be 1 when  $x = e = f = 1$ . That is  $e^* = e \bar{f} \vee x \bar{e} f \vee x e f = e \bar{f} \vee x f$ .  $y = \bar{x} \bar{e} f \vee x \bar{e} f \vee x e f = \bar{x} f \vee x \bar{e} \bar{f}$ .



We terminate this discussion with the schematic diagram above. It should be clear however, how to proceed to construct a detailed net.

## 9. BIBLIOGRAPHY

### Books on Switching and Automata Theory

1. Shannon, C., and McCarthy, J., eds. Automata Studies, Princeton Univ. Press, 1956.
2. Caldwell, S. H., Switching Circuits and Logical Design, John Wiley and Sons, Inc., New York, 1958.
3. Keister, W., Ritchie, A. E., and Washburn, S. H., The Design of Switching Circuits, D. Van Nostrand Co., New York, 1951.
4. Phister, M., Jr., Logical Design of Digital Computers, John Wiley and Sons, Inc., New York, 1958.
5. Richards, R. K., Arithmetic Operations in Digital Computers, D. Van Nostrand, Co., New York, 1955.
6. Richards, R. K., Digital Computer Components and Circuits, D. Van Nostrand, Co., New York, 1957.
7. Seshu, S., Theory of Linear Graphs with Application to Electrical Engineering, Electrical Engineering Department, Syracuse Univ., Syracuse, New York.
8. Staff of Harvard Computation Laboratory, Synthesis of Electronic Computing and Control Circuits, Harvard Univ. Press, Cambridge, 1951.

### Books on Related Mathematical Topics

9. Birkhoff, G., Lattice Theory, American Mathematical Society Colloquium Publications, Vol. XXV, (1948).
10. Birkhoff, G., and Mac Lane, S., A Survey of Modern Algebra, The Macmillan Co., New York, (Chapter XI).
11. Church, A., Introduction to Mathematical Logic, Vol. I, Princeton Univ. Press, Princeton, 1956.
12. Davis, M., Computability and Unsolvability, McGraw-Hill, New York, 1958.
13. Hilbert, D., and Ackermann, W., Principles of Mathematical Logic, Chelsea, New York, 1950 (translation of the 2nd edition, 1928).
14. Kleene, S. C., Introduction to Metamathematics, D. Van Nostrand Co., Princeton, New Jersey, 1952.

15. Rosenbloom, P. C., The Elements of Mathematical Logic, Dover Publications, New York, 1950.
16. van der Waerden, B. L., Modern Algebra, Vols. I, II, Ungar, New York, 1949-50 (translation of the 2nd edition, 1937, 1940).

#### Papers and Articles

17. Bratton, D., "A Theorem on Factorization in Semi-groups that Results from a Theorem of Kleene's in the Theory of Automata," Abstract in Notices of the American Mathematical Society, Vol. 6, No. 3 (June, 1958).
18. Büchi, J. R., Elgot, C. C., and Wright, J. B., "The Non-existence of Certain Algorithms of Finite Automata Theory," Abstract in Notices of the American Mathematical Society, Vol. 5, No. 1, 98 (February, 1958).
19. Burks, A. W., "The Logic of Fixed and Growing Automata," to appear in Proceedings of the 1957 International Symposium on Switching Theory, Harvard University Press.
20. Burks, A. W., and Copi, I. M., "The Logical Design of an Idealized General-Purpose Computer," J. Franklin Institute, 261, No. 3, 299-314 (1956).
21. Burks, A. W., McNaughton, R., Pollmar, C. H., Warren, D. W., and Wright, J. B., "Complete Decoding Nets: General Theory and Minimality," J. Soc. Ind. Appl. Math., 2, 201-243 (1954).
22. Burks, A. W., McNaughton, R., Pollmar, C. H., Warren, D. W., and Wright, J. B., "The Folded Tree," J. of the Franklin Institute, 260, Nos. 1 and 2, 7-24 and 115-126 (1955).
23. Burks, A. W., and Wang, H., "The Logic of Automata," J. of the Assoc. for Computing Machinery, Part I, 4, 193-218, Part II, 4, 279-297 (1957).
24. Burks, A. W., Warren, D. W., and Wright, J. B., "An Analysis of a Logical Machine Using Parenthesis-Free Notation," Mathematical Tables and Other Aids to Computation (April, 1954).
25. Burks, A. W., and Wright, J. B., "Theory of Logical Nets," Proc. IRE, 41, 1357-1365 (1953).
26. Church, A., "Application of Recursive Arithmetic in the Theory of Computers and Automata," Notes from "Advanced Theory of the Logical Design of Digital Computers," Summer Session Course, U. of Michigan, June, 1958.
27. Copi, I. M., "Matric Development of the Calculus of Relations," J. of Symbolic Logic, 13, 193-203.

28. Copi, I. M., Elgot, C. C., and Wright, J. B., "Realization of Events by Logical Nets," J. of the Assoc. for Computing Machinery, 5, 181-196 (1958).
29. Elgot, C., "On Single vs. Triple Address Computing Machines," J. of the Assoc. for Computing Machinery, 1, 119-123 (1954).
30. Elgot, C., and Wright, J., "Series-Parallel Graphs and Lattices," to appear in the Duke Mathematical Journal, 1958.
31. Elgot, C., and Wright, J., "Quantifier Elimination in a Problem of Logical Design," to appear in the Michigan Mathematical Journal.
32. Friedman, J., "Some Results in Church's Restricted Recursive Arithmetic," J. of Symbolic Logic, 22, 337-342 (1957).
33. Ginsberg, S., "On the Length of the Smallest Uniform Experiment which Distinguishes the Terminal States of a Machine," J. of the Assoc. for Computing Machinery, 5, 266-280 (1958).
34. Ginsberg, S., "Some Remarks on Abstract Machines," I, January, 1958, II, May, 1958, National Cash Register Co., Electronics, Div., 1401 El Segundo Blvd., Hawthorne, Calif.
35. Hohn, F. E., Seshu, S., and Aufenkamp, D. D., "The Theory of Nets," IRE Transactions on Electronic Computers (September, 1957).
36. Huffman, D. A., "The Synthesis of Sequential Switching Circuits," J. of the Franklin Institute, 257, 161-190, 275-303 (1954).
37. Kleene, S. C., "Representation of Events in Nerve Nets and Finite Automata," Automata Studies, Princeton Univ. Press, 1956, 3-41.
38. DeLeeuw, K., Moore, E. F., Shannon, C. E., and Shapiro, N., "Computability by Probabilistic Machines," Automata Studies, Princeton Univ. Press, 1956, 183-212.
39. Lunts, A. G., "The Application of Boolean Matrix Algebra to the Analysis and Synthesis of Relay Contact Networks," Doklady Akademii Nauk SSSR, 70, No. 3
40. McCluskey, E. J., "Iterative Combinational Switching Networks," Notes from "Advanced Theory of The Logical Design of Digital Computers," Summer Session Course, U. of Michigan, June, 1958.
41. McCluskey, E. J., Jr., "Minimization of Boolean Functions," Bell System Technical Journal, 35, 1417-1444 (1956).
42. McNaughton, R., "A Proof that Addition is Not Arithmetically Definable in Terms of a Single Unary Operator," Technical Report No. 3, Applied Mathematics and Statistics Laboratory, Stanford University, May 1, 1957.

43. McNaughton, R., "unate Truth Functions," Technical Report No. 4, Stanford University, October 21, 1957.
44. McNaughton, R., and Mitchess, B., "The Minimality of Rectifier Nets with Multiple Outputs Incompletely Specified," Technical Report No. 2, Applied Mathematics and Statistics Laboratory, Stanford University, April 18, 1957.
45. Mealy, G. H., "A Method for Synthesizing Sequential Circuits," Bell System Technical Journal, 34, 1045-1079 (1955).
46. Medvedev, I. T., "On a Class of Events Representable in a Finite Automaton," MIT Lincoln Lab Group Report, 34-73, translated from the Russian by J. Schorr-Kon, June 30, 1958.
47. Moore, E. F., "A Simplified University Turing Machine," Proc. of the Toronto Meeting of the ACM, Toronto, 1952 (Bell Telephone Lab. monograph 2098).
48. Moore, E. F., "Gedanken-Experiments on Sequential Machines," Automata Studies, Princeton, 1956, 129-153.
49. Muller, D. E., "A Theory of Asynchronous Circuits," to appear in Proceedings of the 1957 International Symposium on Switching Theory, Harvard Univ. Press.
50. Murray, F. J., "Mechanisms and Robots," J. of the Assoc. for Computing Machinery, 2, 61-82 (1955).
51. Polya, G., "Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und Chemische Verbindungen," Acta Mathematica, 68, 145-254 (1937).
52. Post, E. L., "Revursively Enumerable Sets of Positive Integers and Their Decision Problems," Bull. Amer. Math. Soc., 50, 284-316 (1944).
53. Quine, W. V., "A Way to Simplify Truth Functions," American Mathematical Monthly, 62, 627-631 (1955).
54. Rabin, M., and Scott, D., "Finite Automata and their Decision Problems," IBM Research Center, Lamb Estate, Yorktown, New York, 1958).
55. Raney, G. N., "Sequential Functions," J. of the Association for Computing Machinery, 5, 177-180 (1958).
56. Riordan, J., and Shannon, C. E., "The Number of 2-Terminal Series-Parallel Networks," J. of Mathematics and Physics, 21, 83 ff. (1942).
57. Shannon, C. E., "A Universal Turing Machine with Two Internal States," Automata Studies, Princeton Univ. Press, 1956, 157-165.



58. Shannon, C. E., "Symbolic Analysis of Relay and Switching Circuits," AIEE Trans., 57, 713-723 (1938).
59. Shannon, C. E., "The Synthesis of Two-Terminal Switching Circuits," Bell System Technical Journal, 28, No. 1, 59-98 (January, 1949).
60. Turing, A. M., "On Computable Numbers, with an Application to the Entscheidungsproblem," Proc. London Math. Soc. ser. 2, 42, 230-265 (1936-7), with a correction, ibid., 43, 544-546 (1937).
61. Wang, H., "A Variant to Turing's Theory of Computing Machines," J. of the Assoc. for Computing Machinery, 4, 63-92 (1957).
62. Wang, H., "University Turing Machines: An Exercise in Coding," Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 3, 69-80 (1957).

UNIVERSITY OF MICHIGAN



3 9015 02826 7618